

**RANCANG BANGUN APLIKASI *MOBILE LEARNING*
CLIENT SERVER BERBASIS *MOODLE* PADA PLATFORM
*ANDROID***

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik Pada
Jurusan Teknik Informatika

Oleh :

THIO PRATAMA PUTRA
NIM : 10751000091



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SULTAN SYARIF KASIM RIAU
PEKANBARU
2011**

APPLICATION DESIGN MOBILE LEARNING CLIENT SERVER BASIC OF MOODLE ON ANDROID PLATFORM

THIO PRATAMA PUTRA

10751000091

Final Exam Date: November 1 8th, 2011

Graduation Ceremony Period: February 2012

Information Engineering Department

Faculty of Sciences and Technology

State Islamic University of Sultan Syarif Kasim Riau

ABSTRACT

Convergence technology communication device makes communication on device functions has exceeded its basic functions nowadays. Smartphone is a device that has implemented the convergence of communication technology devices, such as the implementation of the operating system on the device. Android is Google's developed operating system based on Open Source. This research will build a mobile learning client server application. This application run on the Android operating system which is integrated with the Moodle Learning System. The system used the Java programming language. The system run in Android devices intergrated with Moodle connected by connecting engine built by PHP programming language. This machine handles the request and respone from client to server and the opposite. Result from the test performed on constructed system showed that the system has been running well and as expected for further development. For further development is expected to have additional features. It is to be Expected by the application of mobile learning will be able to assist in the learning process.

Key words: *Android, Mobile Learning, Moodle.*

RANCANG BANGUN APLIKASI *MOBILE LEARNING CLIENT* *SERVER* BERBASIS *MOODLE* PADA *PLATFORM ANDROID*

THIO PRATAMA PUTRA

10751000091

Tanggal Sidang: 18 November 2011

Periode Wisuda: Februari 2012

Jurusan Teknik Informatika

Fakultas Sains dan Teknologi

Universitas Islam Negeri Sultan Syarif Kasim Riau

ABSTRAK

Konvergensi teknologi perangkat komunikasi pada saat ini membuat fungsi perangkat komunikasi telah melebihi fungsi dasarnya. *Smartphone* merupakan perangkat yang telah menerapkan konvergensi teknologi perangkat komunikasi, seperti penerapan sistem operasi di dalam perangkat tersebut. *Android* merupakan sistem operasi buatan Google yang sedang berkembang dan *open source*. Pada penelitian ini akan dibangun sebuah aplikasi *mobile learning client server* yang bisa berjalan di sistem operasi *Android* yang terintegrasi dengan *Learning Management System Moodle*. Sistem yang dibangun menggunakan bahasa pemrograman Java. Antara sistem yang berjalan di perangkat *Android* dengan sistem yang ada di *Moodle* dijematani oleh mesin penghubung yang dibangun menggunakan bahasa pemrograman PHP. Mesin ini menangani *request* dan *response* dari *client* ke *server* dan sebaliknya. Hasil dari pengujian yang dilakukan terhadap sistem yang dibangun menunjukkan sistem telah berjalan dengan baik dan sesuai yang diharapkan. Untuk pengembangan lebih lanjut diharapkan ada penambahan fitur. Diharapkan dengan adanya aplikasi *mobile learning* ini akan dapat membantu dalam proses pembelajaran.

Kata kunci: *Android, Mobile Learning, Moodle.*

DAFTAR ISI

HALAMAN JUDUL LAPORAN.....	i
LEMBAR PERSETUJUAN	ii
LEMBAR PENGESAHAN	iii
LEMBAR HAK ATAS KEKAYAAN INTELEKTUAL	iv
LEMBAR PERNYATAAN	v
LEMBAR PERSEMBAHAN	vi
ABSTRACT	vii
ABSTRAK	viii
KATA PENGANTAR.....	ix
DAFTAR ISI	xi
DAFTAR GAMBAR	xv
DAFTAR LAMPIRAN	xvii
DAFTAR TABEL	xviii
DAFTAR SIMBOL	xix
DAFTAR ISTILAH	xxi
BAB I PENDAHULUAN.....	I-1
1.1. Latar Belakang	I-1
1.2. Rumusan Masalah	I-3
1.3. Batasan Masalah.....	I-3
1.4. Tujuan Penelitian	I-3
1.5. Sistematika Penulisan	I-4
BAB II LANDASAN TEORI	II-1
2.1. <i>E-Learning</i>	II-1
2.1.1. Sejarah <i>e-Learning</i>	II-1
2.1.2. Keuntungan <i>e-Learning</i>	II-2
2.1.3. Model <i>e-Learning</i>	II-3

2.1.4. Kategori <i>e-Learning</i>	II-4
2.2. <i>Mobile Learning</i>	II-5
2.2.1. Sejarah <i>Mobile Learning</i>	II-5
2.2.2. Pengertian <i>Mobile Learning</i>	II-6
2.2.3. Konsep <i>Mobile Learning</i>	II-8
2.2.4. Keuntungan <i>Mobile Learning</i>	II-8
2.2.5. Keterbatasan <i>Mobile Learning</i>	II-9
2.3. <i>Moodle</i>	II-10
2.3.1. Fitur <i>Moodle</i>	II-11
2.3.2. Karakteristik Penggunaan <i>Moodle</i>	II-12
2.4. Konvergensi Perangkat <i>Mobile</i>	II-12
2.5. <i>Android</i>	II-13
2.5.1. Sejarah <i>Android</i>	II-13
2.5.2. Arsitektur <i>Android</i>	II-14
2.5.2.1. <i>Linux kernel</i>	II-15
2.5.2.2. <i>Libraries</i>	II-15
2.5.2.3. <i>Android Runtime</i>	II-16
2.5.2.4. <i>Applications Framework</i>	II-16
2.5.2.5. <i>Applications dan Widgets</i>	II-17
2.5.3. <i>Android SDK (Software Development Kit)</i>	II-17
2.5.4. Komponen Aplikasi	II-17
2.5.3. Kelebihan <i>Platform Android</i>	II-19
2.6. Analisa dan Perancangan Berorientasi Objek	II-20
2.6.1. <i>Unified Modelling Language</i>	II-20
2.6.1.1. <i>Use Case Diagram</i>	II-21
2.6.1.2. <i>Class Diagram</i>	II-21
2.6.1.3. <i>Sequence Diagram</i>	II-21
2.6.1.4. <i>Deployment Diagram</i>	II-22
2.6.1.5. <i>Activity Diagram</i>	II-22

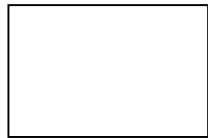
2.7. <i>Rational Unified Process</i>	II-22
2.7.1. Pengertian RUP	II-23
2.7.2. Fase RUP	II-25
BAB III METODOLOGI PENELITIAN	III-1
3.1. Tahapan Penelitian	III-1
3.2. Tahapan RUP	III-2
3.2.1. Fase <i>Inception</i>	III-2
3.2.2. Fase <i>Elaboration</i>	III-2
3.2.3. Fase <i>Contruction</i>	III-3
3.2.4. Fase <i>Transtition</i>	III-3
BAB IV ANALISA DAN PERANCANGAN	IV-1
4.1. Deskripsi Arsitektur Sistem	IV-1
4.2. Deskripsi Umum Kebutuhan Sistem	IV-3
4.2.1. Sistem Yang Akan Dibangun	IV-3
4.2.1.1. Sistem Pada Perangkat Android	IV-3
4.2.1.2. Mesin Penghubung	IV-4
4.3. Fungsi Sistem	IV-5
4.3.1. Fungsi Sistem dari Sisi Perangkat Android	IV-5
4.3.2. Fungsi Mesin Penghubung	IV-6
4.4. Karakteristik Pengguna	IV-6
4.5. Perancangan Sistem	IV-7
4.5.1. Pemodelan UML (Unified Modelling Language)	IV-8
4.5.1.1. <i>Use Case Diagram</i>	IV-8
4.5.1.2. <i>Class Diagram</i>	IV-11
4.5.1.3. <i>Activity Diagram</i>	IV-15
4.5.1.4. <i>Sequence Diagram</i>	IV-16
4.5.1.5. <i>Deployment Diagram</i>	IV-17
4.5.2. Hubungan Class di Android denga Mesin Penghubung...	IV-17
4.5.3. Struktur Menu Sistem	IV-18

4.5.4. Perancangan Antar Muka Sistem	IV-18
4.5.4.1 Perancangan Antar Muka Login di Android	IV-18
BAB V IMPLEMENTASI DAN PENGUJIAN.....	V-1
5.1. Implementasi	V-1
5.1.1. Lingkungan Pengembangan	V-1
5.1.2. Lingkungan Implementasi.....	V-2
5.1.3. Tahap-tahap Implementasi	V-2
5.1.3.1. Instalasi Moodle	V-2
5.1.3.2. Konfigurasi Awal	V-2
5.1.3.3. Instalasi Mesin Penghubung ke <i>Hosting</i>	V-4
5.1.3.4. Instalasi aplikasi <i>Mobile Learning</i>	V-6
5.1.4. Implementasi Unjuk Kerja pada Perangkat Android	V-6
5.2. Pengujian Sistem	V-7
5.2.1. Pengujian Akses Aplikasi <i>Mobile Learning</i>	V-8
5.2.2. Pengujian <i>Blackbox</i> Aplikasi <i>Mobile Learning</i>	V-9
5.2.3. Kesimpulan Pengujian.....	V-14
BAB VI PENUTUP	VI-1
6.1. Kesimpulan.....	VI-1
6.2. Saran.....	VI-1
DAFTAR PUSTAKA	
LAMPIRAN	
RIWAYAT HIDUP	

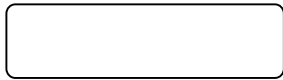
DAFTAR TABEL

Tabel	Halaman
4.1. Tabel Deskripsi pengguna.....	IV-6
4.2. Spesifikasi <i>Use Case login</i>	IV-9
4.3. Spesifikasi <i>Use Case</i> menampilkan <i>course</i>	IV-9
4.4. Spesifikasi <i>Use Case</i> menampilkan tugas.....	IV-9
4.5. Spesifikasi <i>Use Case upload</i> tugas	IV-9
4.6. Spesifikasi <i>Use Case</i> menampilkan nilai	IV-10
4.7. Spesifikasi <i>Use Case download file</i>	IV-10
4.8. Spesifikasi <i>Use Case</i> menampilkan berita	IV-10
4.9. Detail <i>class diagram</i>	IV-12
4.10. Deskripsi antar muka <i>login</i> pada perangkat <i>Android</i>	IV-20
5.1. Pengujian akses ke aplikasi <i>mobile learning</i> dari <i>device</i> <i>Android</i>	V-8
5.2. Pengujian aplikasi dengan metode <i>Blackbox</i>	V-11
B.1. Deskripsi antar muka <i>login</i>	B-1
B.2. Deskripsi antar muka halaman utama.....	B-2
B.3. Deskripsi antar muka halaman <i>course</i>	B-3
B.4. Deskripsi antar muka halaman tugas	B-4
B.5. Deskripsi antar muka halaman nilai	B-5
B.6. Deskripsi antar muka halaman <i>download file</i>	B-6
B.7. Deskripsi antar muka berita.....	B-7

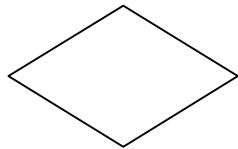
DAFTAR SIMBOL



Proses pada *flowchart*



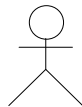
Start/ Finish suatu proses pada *flowchart*



Kondisi/ keadaan tindakan pada *flowchart* berupa Ya atau Tidak



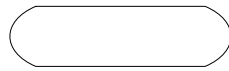
Alur/ langkah pada *flowchart* dan model data spasial



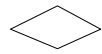
Pengguna Sistem (Aktor)



Proses (Use case)



Aktivitas (Activity)



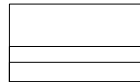
Keputusan (decision)



Status mulai (start state)



Status selesai (end state)



Kelas (class)

DAFTAR ISTILAH

<i>Android</i>	: Sistem operasi untuk perangkat komunikasi jenis smartphone.
<i>Android SDK</i>	: Kumpulan library untuk development aplikasi android.
<i>Android Runtime</i>	: Pustaka inti penyedia fungsi-fungsi di Android.
<i>Client</i>	: pengguna sistem.
<i>Database</i>	: basis data yang digunakan untuk menyimpan data.
<i>Interface</i>	: antarmuka.
<i>Java development kit</i>	: Merupakan kumpulan library untuk pemrograman Java
<i>Linux kernel</i>	: Inti sistem operasi linux.
<i>Mobile learning</i>	: Sistem pembelajaran bergerak yang dilakukan di perangkat mobile.
<i>Moodle</i>	: Merupakan sebuah perangkat lunak yang berfungsi sebagai alat bantu sistem pembelajaran elektronik
<i>Server</i>	: yang menyediakan layanan sistem.
<i>Smartphone</i>	: Perangkat komunikasi yang didalamnya telah ditanamkan sistem pintar seperti sistem operasi dan aplikasi pendukung lainnya.
<i>Web browser</i>	: aplikasi untuk menjelajahi informasi berbasis <i>web</i> .

BAB I

PENDAHULUAN

1.1. Latar Belakang

Perkembangan perangkat komunikasi telepon selular belakangan ini mengalami pertumbuhan yang sangat pesat yang didukung dengan disertakannya teknologi terbaru dalam perangkat tersebut. Perkembangan itu juga ditambah dengan konvergensi perangkat komunikasi yang mampu menangani proses *voice*, *data* dan *video* dalam satu perangkat. Berbagai macam fitur pendukung juga disisipkan pada perangkat ini, seperti pengolah dokumen, gambar, video, konektifitas dan sebagainya.

Berdasarkan data ATSI (Asosiasi Telepon Seluler Indonesia) 14 Juli 2010, 180 juta lebih penduduk Indonesia telah menjadi pelanggan layanan seluler, itu berarti 180 juta orang juga memiliki perangkat telepon seluler. Lebih dari 30 persennya aktif menggunakan layanan *broadband* dari layanan seluler tersebut.

Begitu banyak perangkat *mobile* yang digunakan untuk mendukung kegiatan manusia, salah satunya adalah di bidang pendidikan. Untuk saat ini, setidaknya institusi pendidikan haruslah mampu memperbaiki sistem yang ada dalam bentuk *improving teaching-learning process* (perubahan dalam proses belajar mengajar). Kenyataan ini menjadi sebuah peluang bagi institusi pendidikan untuk mencoba menyelenggarakan pembelajaran memanfaatkan media bergerak atau yang lebih dikenal dengan *mobile learning*.

Mobile learning adalah suatu pendekatan pembelajaran yang melibatkan perangkat bergerak seperti telepon genggam, PDA, dan *tablet PC*, dimana pembelajar dapat mengakses materi, arahan dan aplikasi yang berkaitan dengan pelajaran tanpa dibatasi oleh ruang dan waktu, dimanapun dan kapanpun mereka berada.

Terkait dengan jumlah pengguna perangkat *mobile* yang besar. *Mobile learning* dapat dijadikan salah satu alternatif untuk memecahkan masalah dalam dunia pendidikan, diantaranya masalah pemerataan akses terhadap konten pendidikan, kualitas konten dan lainnya. Kemudian memperkuat sumber informasi pengguna *mobile learning* dan menekan biaya terhadap akses konten pendidikan tersebut.

Studi serta tindak lanjut terhadap *mobile learning* ini telah banyak dilakukan baik itu dari pemerintah maupun swasta. Salah satu yang pernah dilakukan oleh Yayasan Telematika Edukasi Indonesia adalah membangun M-Learning sebagai sumber belajar berbasis WAP (*Wireless Application Protocol*). Selain itu jurnal-jurnal yang membahas *mobile learning* ini telah banyak yang dipublikasikan sebagai contoh, David Parsons, Hakyoungh Ryu & Mark Cranshaw (2007) membahas tentang *A Design Requirements Framework for Mobile Learning Environment*, dan masih banyak lagi jurnal yang membahas tentang *mobile learning*.

Namun demikian, studi dan tindak lanjut yang dilakukan masih sering menemukan halangan, diantaranya yang paling umum adalah masalah biaya dari pemerintah yang tidak merata, sulitnya merubah paradigma dari belajar konvensional ke belajar dengan model *mobile learning*, dan keterbatasan sumber daya manusia.

Inovasi baru perlu dilakukan sesuai perkembangan teknologi yang mendukung dari *mobile learning* itu sendiri. Salah satu inovasinya adalah pengembangan dari sisi perangkat lunak dan mengintegrasikan dengan salah satu *Learning Management System* (LMS) seperti LMS Moodle, dengan sistem *client server*. Moodle merupakan LMS yang banyak dipakai oleh institusi pendidikan di Indonesia. Moodle adalah sebuah perangkat lunak *open source* yang berfungsi sebagai *Learning Management system* untuk *e-learning*.

Kemudian inovasi disisi perangkat lunak juga perlu dilakukan, seperti aplikasi yang bisa berjalan pada sistem operasi terbaru untuk *smartphone*, *PDA*, *tablet PC* atau perangkat *mobile* pada saat ini, salah satunya seperti *Android*. *Android* merupakan sebuah sistem operasi untuk perangkat *mobile*, yang dibangun

diasas *Linux Kernel* versi 2.6, dimana *kernel* ini telah teruji ketangguhannya di berbagai *distro Linux*. *Android* sendiri merupakan generasi baru *platform mobile*, *platform* ini bersifat *open source* sehingga siapapun bebas mengubah *source code* sesuai kebutuhan. *Android* bertujuan untuk memajukan piranti lunak agar setiap pengguna bisa mendapatkan pengalaman baru serta menggali potensi yang ada *platform* ini.

Dengan berbagai penjelasan diatas diharapkan riset ini nantinya bisa meningkatkan minat pelajar menggunakan *mobile learning*. Kemudian *mobile learning* ini bisa di integrasi dengan *moodle* sebagai *Learning Management System* yang sudah banyak dipakai di institusi pendidikan Indonesia, sehingga menghasilkan produk yang bermanfaat.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah diatas, yang menjadi pokok permasalahan dalam hal ini adalah “Bagaimana merancang dan membangun aplikasi *mobile learning client server* berbasis *moodle* pada *platform android*”.

1.3 Batasan Masalah

Untuk mendapatkan hasil yang optimal, maka akan diberikan batasan-batasan masalah dalam penulisan Tugas Akhir ini, agar tidak jauh keluar dari pembahasan. Tugas Akhir ini hanya dibatasi sebagai berikut:

1. Riset ini hanya membahas fungsi *Moodle* dalam *mobile learning client server* yang akan berjalan pada *platform Android*.
2. Sistem yang akan dibangun tidak membahas aspek-aspek keamanan dari sisi *client server*. Pada *server*, data *directory moodle* yang bisa diakses secara langsung karena *.htaccess* yang di non-aktifkan. Sedangkan pada *client* tidak menggunakan *session*.
3. Sistem yang berjalan pada *platform Android* ditujukan untuk *student*.

1.4 Tujuan Penelitian

Adapun tujuan yang ingin dicapai penulis dari Tugas Akhir ini adalah untuk membangun aplikasi *mobile learning client server* berbasis *moodle* yang bisa berjalan pada *platform Android*.

1.5 Sistematika Penulisan

Berikut merupakan rencana susunan sistematika penulisan laporan Tugas Akhir yang akan dibuat :

Bab I Pendahuluan

Bab ini berisi penjelasan mengenai latar belakang masalah, rumusan masalah, batasan masalah, tujuan dan sistematika penulisan dari Tugas Akhir yang dibuat.

Bab II Landasan Teori

Bab ini membahas tentang teori-teori umum dan khusus yang berhubungan dengan tugas akhir ini.

Bab III Metodologi Penelitian

Bab ini membahas langkah-langkah yang dilaksanakan dalam proses penelitian dengan metode *Rational Unified Process* (RUP), yaitu tahapan *inception*, *elaboration*, *contruction*, dan *transition*.

Bab IV Analisa dan Perancangan

Bab ini berisi pembahasan mengenai kebutuhan sistem, yang terdiri dari : *Flowchart system*, *UML*, *User interface*, perancangan menggunakan pendekatan berorientasi objek.

Bab V Implementasi dan Pengujian

Bab ini berisi penjelasan mengenai implementasi yang terdiri dari: batasan implementasi, lingkungan implementasi, hasil implementasi, pengujian sistem dan kesimpulan pengujian

Bab VI Kesimpulan dan Saran

Bagian ini berisi kesimpulan yang dihasilkan dari pembahasan tentang pengembangan aplikasi *mobile learning* dijalankan pada *platform Android*.

BAB II

LANDASAN TEORI

2.1 *E-Learning*

Definisi *Electronic learning (E-Learning)* mengandung istilah yang begitu luas, sehingga banyak pakar yang menguraikan arti *e-Learning* tersebut. Beberapa definisi dari *e-Learning* diantaranya adalah:

e-Learning merupakan suatu jenis belajar mengajar yang memungkinkan tersampainya bahan ajar ke siswa dengan menggunakan media Internet, Intranet atau media jaringan komputer lain. (Hartley, 2001) dalam Wahono (2008)

e-Learning adalah sistem pendidikan yang menggunakan aplikasi elektronik untuk mendukung belajar mengajar dengan media Internet, jaringan komputer, maupun komputer standalone. Glossary of *e-Learning* Terms (Glossary, 2001) dalam Wahono (2008)

Dari beberapa definisi diatas dapat disimpulkan bahwa *e-Learning* merupakan suatu sistem pendidikan yang memanfaatkan teknologi informasi dalam proses belajar mengajar.

2.1.1 Sejarah *e-Learning*

Uraian singkat tentang perkembangan *e-Learning* dari masa ke masa adalah seperti di bawah (Cross, 2002) dalam Wahono (2008):

1. 1990: CBT (*Computer Based Training*)

Era dimana mulai bermunculan aplikasi *e-Learning* yang berjalan dalam *PC standalone* ataupun berbentuk kemasan CD-ROM. Isi berupa materi dalam bentuk tulisan maupun multimedia (*video dan audio*) dalam format MOV, MPEG-1 atau AVI. Perusahaan perangkat lunak *Macromedia* mengeluarkan tool pengembangan bernama *Authorware*, sedangkan *Asymetrix* (sekarang bernama Click2learn) juga mengembangkan perangkat lunak bernama *Toolbook*.

2. 1994: Paket-Paket CBT

Seiring dengan mulai diterimanya CBT oleh masyarakat, sejak tahun 1994 muncul CBT dalam bentuk paket-paket yang lebih menarik dan diproduksi secara massal.

3. 1997: LMS (*Learning Management System*)

Seiring dengan perkembangan teknologi internet di dunia, masyarakat dunia mulai terkoneksi dengan internet. Kebutuhan akan informasi yang cepat diperoleh menjadi mutlak, dan jarak serta lokasi bukanlah halangan lagi. Disinilah muncul sebutan *Learning Management System*

atau biasa disingkat dengan LMS. Perkembangan LMS yang semakin pesat membuat pemikiran baru untuk mengatasi masalah interoperability antar LMS yang ada dengan suatu standard. Standard yang muncul misalnya adalah standard yang dikeluarkan oleh AICC (*Airline Industry CBT Committee*), IMS, IEEE LOM, ARIADNE, dsb.

4. 1999: Aplikasi *e-Learning* Berbasis Web

Perkembangan LMS menuju ke aplikasi *e-Learning* berbasis *Web* secara total, baik untuk pembelajar (*learner*) maupun administrasi belajar mengajarnya. LMS mulai digabungkan dengan situs-situs portal yang pada saat ini boleh dikata menjadi rujukan situs-situs informasi, majalah, dan surat kabar dunia. Isi juga semakin kaya dengan berpaduan *multimedia*, *video streaming*, serta penampilan interaktif dalam berbagai pilihan format data yang lebih standar, berukuran kecil dan stabil.

2.1.2 Keuntungan *e-Learning*

Ada beberapa keuntungan yang diperoleh dalam pemanfaatan *e-Learning* diantaranya (Wahono, 2008):

1. Menghemat waktu proses belajar mengajar
2. Mengurangi biaya perjalanan
3. Menghemat biaya pendidikan secara keseluruhan (infrastruktur, peralatan, buku-buku)
4. Menjangkau wilayah geografis yang lebih luas
5. Melatih pembelajar lebih mandiri dalam mendapatkan ilmu pengetahuan

2.1.3 Model *e-Learning*

Dengan adanya perkembangan teknologi informasi maka model pembelajaran untuk *e-Learning* dapat dibagi dalam beberapa model, berikut model pembelajaran tersebut (Widhiartha, 2008):

1. *Computer Based Learning/Training* (CBL/ CBT)

Model CBL/CBT berkembang sekitar pertengahan tahun 1990-an. Saat itu berbagai pelatihan atau kelas menyediakan berbagai bahan belajar berupa modul elektronik baik berupa perangkat lunak edukasi maupun *softcopy* dari berbagai modul cetak yang sudah ada sebelumnya. Bentuk ini di kemudian hari dikenal sebagai *e-book* dan berkembang semakin pesat berkat adanya format file pdf dari *Adobe*. Pada era tersebut CBL/CBT sendiri berkembang pada komputer *stand alone* dan belum terhubung dengan *internet*. Biasanya pembelajaran dengan model CBL/CBT adalah untuk penyiapan tenaga ahli pada suatu bidang yang memerlukan pelatihan terlebih dahulu sebelum menempati posisinya. Perangkat lunak simulasi membantu peserta didik melakukan simulasi atas pekerjaan yang hendak dilakukan. Dengan simulasi maka proses belajar menjadi lebih mudah dan biaya pun bisa ditekan lebih murah dibandingkan apabila mereka harus mempraktekkan sendiri pada peralatan yang sebenarnya. Modul elektronik mempermudah peserta untuk mempelajari secara mandiri materi yang harus dipelajari dan tidak memerlukan biaya cetak yang tinggi.

2. *Web-based Learning*

Berbagai perangkat lunak edukasi ataupun *softcopy* dari modul, diktat, dan berbagai buku elektronik (e-book) lainnya yang semula didistribusikan dalam bentuk disket atau CD mulai membanjiri *internet*. Dengan melakukan upload berbagai referensi dan bahan belajar di *internet* berarti membuka akses dari seluruh penjuru dunia terhadap berbagai bahan belajar tersebut. Para pengguna *internet* pun bisa mempelajari apa saja dari berbagai situs *web* yang tersedia. Demikian pula para penyelenggara pendidikan mulai memanfaatkan *internet* untuk memperluas layanan mereka pada siapapun yang ingin menjadi peserta didiknya. Berbagai kelas dan pelatihan bisa diikuti hanya dengan

melakukan berbagai *download* terhadap bahan belajar elektronik, berdiskusi dengan dosen melalui email atau forum-forum diskusi *online*, dan mengikuti ujian secara *online* di *internet*. Setelah lulus sang peserta didik tinggal menunggu ijazah atau sertifikat yang terkirim ke alamatnya. Model inilah yang dikenal sebagai *Web-based learning*, sebuah model pembelajaran jarak jauh yang menggunakan internet sebagai sarannya.

1. *Mobile Learning*

E-Learning tidak hanya terbatas pada penggunaan komputer saja. Berbagai model pembelajaran yang menggunakan peralatan elektronik lainnya seperti misalnya telepon genggam pun saat ini telah mulai berkembang. Dengan berbagai fitur dan teknologi yang dimiliki telepon genggam saat ini telah melahirkan sebuah model pembelajaran baru yang dikenal sebagai *mobile learning* (m-learning). Aktivitas utama pada *M-learning* adalah mendistribusikan bahan belajar kepada peserta didik agar dapat diakses menggunakan perangkat komunikasi portabel semacam telepon genggam, *PDA* atau *smartphone*.

2.1.4 Kategori *e-Learning*

E-Learning sendiri memiliki kategori dalam cara pembelajarannya, berikut kategori dari *e-Learning* (Widhiartha, 2008):

1. *Synchronous Learning*

Pada pembelajaran *synchronous* kondisinya mirip dengan pembelajaran konvensional hanya saja pada *e-Learning* hal ini tidak ditandai dengan kehadiran secara fisik. Pada bentuk *synchronous* ini pendidik (instruktur), peserta didik dan rekan-rekannya melakukan “pertemuan” secara *online* di internet. Melakukan proses belajar mengajar seolah sedang berada pada ruang fisik yang sama.

2. *Self-directed Learning*

Pada kategori ini peserta didik melakukan pembelajaran secara mandiri dengan mengakses berbagai referensi dan bahan belajar yang disediakan. Tidak ada instruktur ataupun waktu khusus untuk berdiskusi dengan sesama

peserta didik. Masing-masing peserta didik melakukan proses belajar sesuai dengan kebutuhannya.

3. *Asynchronous Learning*

Kategori ini mengkombinasikan karakteristik dari kedua kategori sebelumnya. Peserta didik belajar secara mandiri namun tetap berkomunikasi dengan peserta didik lainnya maupun dengan pendidik walaupun tidak harus di waktu khusus. Penggunaan *email*, *instant message* (Yahoo! Messenger, Gtalk) ataupun board pada forum dapat digunakan sebagai media komunikasi dan interaksi baik dengan pendidik maupun sesama peserta didik.

2.2 *Mobile Learning*

Mobile learning merupakan pembelejaran bergerak yang memanfaatkan media yang bersifat mobile seperti perangkat komunikasi. Model pembelajaran bergerak ini sudah banyak diterapkan oleh berbagai institusi.

2.2.1 *Sejarah Mobile Learning*

Perkembangan *Mobile learning* tidak lepas dari berkembangnya perangkat-perangkat yang mendukung mobile learning dan studi-studi yang mempelajari sistem *mobile learning* tersebut, berikut periode perkembangan dari *mobile learning*:

1. Perkembangan *Mobile Learning* pada tahun 1970-an dan 1980-an

Terobosan utama dalam penetapan kualitas dan kuantitas terjadi pada tahun 1970 dengan Yayasan Universitas Terbuka Eropa, Open University of the United Kingdom di Milton Keynes, Universidad Nacional de Educación a Distancia di Madrid, Fern Universität di Hagen, Jerman yang diikuti oleh universitas terbuka lainnya di seluruh dunia. Meskipun bahan-bahan kursus universitas terbuka ini menarik banyak perhatian dalam layanan dukungan terhadap siswa karena disediakan universitas-universitas yang memungkinkan untuk menjadi ujung tombak perubahan status pendidikan jarak jauh. Mereka membawa hal tentang itu secara nasional dan diakui secara internasional.

2. Perkembangan *Mobile Learning* pada tahun 1990-an

- a. Universitas-universitas di Eropa dan Asia mengembangkan dan mengevaluasi *mobile learning* untuk para pelajar.
- b. Perusahaan Palm menawarkan bantuan kepada universitas dan perusahaan yang membuat dan menguji penggunaan *mobile learning* pada *platform* PalmOS.
- c. Pembuatan modul *mobile learning* pertama kali untuk sertifikasi CCNA, A+ dan MCSE menggunakan core tools yang kemudian dikenal dengan LMA

3. Perkembangan *Mobile Learning* pada tahun 2000-an

Komisi eropa membiayai proyek multi nasional yaitu MOBIlearn dan M-learning. Perusahaan-perusahaan dibentuk sesuai dengan 3 area *mobile learning* yaitu :

1. *Authoring dan Publishing*
2. *Delivery dan Tracking*
3. *Content Development*

Konferensi dan pameran diadakan untuk memperkenalkan *mobile learning* dan *handheld education*, diantaranya : mLearn, WMUTE, IADIS Mobile Learning. Konferensi bertaraf Internasional diantaranya : ICML di Yordania, Mobile Learning di Malaysia, Handheld Learning di London, dan SALT Mobile di Amerika Serikat.

2.2.2 Pengertian *Mobile Learning*

Mobile Learning merupakan sebuah istilah yang mengacu kepada suatu kegiatan, biasanya pendidikan atau pelatihan yang menggunakan beberapa jenis perangkat *mobile* (David Parson, 2007). Hal ini merupakan upaya dari pemanfaatan teknologi dari perangkat bergerak yang telah melebihi fungsi dasarnya yaitu untuk berkomunikasi. Selain itu perangkat bergerak ini juga telah mendapat dukungan dari segi konektifitasnya.

Mobile Learning sendiri merupakan bagian dari *e-Learning* (*Electronic Learning*), secara konsep *e-Learning* dan *mobile Learning* merupakan bagian dari *d-learning* (*Distance Learning*).

Mobile learning didefinisikan oleh Clark Quinn (Quinn, 2000) sebagai : *The intersection of mobile computing and e-Learning: accessible resources wherever you are, strong search capabilities, rich interaction, powerful support for effective learning, and performance-based assessment. E-Learning independent of location in time or space*. Berdasarkan definisi tersebut, *mobile learning* merupakan model pembelajaran yang memanfaatkan teknologi informasi dan komunikasi. Pada konsep pembelajaran tersebut *mobile learning* membawa manfaat ketersediaan materi ajar yang dapat di akses setiap saat dan visualisasi materi yang menarik.

Untuk lebih memaksimalkan *mobile learning* ada beberapa kemampuan yang harus disediakan oleh perangkat pembelajaran *mobile learning* tersebut seperti kemampuan untuk terkoneksi dengan perangkat lain terutama komputer, kemampuan menyajikan informasi pembelajaran dan kemampuan untuk merealisasikan komunikasi bilateral antara pengajar dan pembelajar. *Mobile learning* adalah pembelajaran yang unik karena pembelajar dapat mengakses materi pembelajaran, arahan dan aplikasi yang berkaitan dengan pembelajaran, kapanpun dan dimanapun. Hal ini akan meningkatkan perhatian pada materi pembelajaran, membuat pembelajaran menjadi pervasif, dan dapat mendorong motivasi pembelajar kepada pembelajaran sepanjang hayat. Selain itu, dibandingkan pembelajaran konvensional, *mobile learning* memungkinkan adanya lebih banyak kesempatan untuk berinteraksi secara informal diantara pembelajar.

Karena *mobile learning* merupakan bagian dari *e-Learning*, maka ada 2 tipe pembagian dari metode pembelajaran ini yaitu, (Efendi dan Zhuang, 2005):

1. *Synchronous training*

Synchronous berarti “pada waktu yang sama”. Jadi *Synchronous training* adalah tipe pelatihan yang terjadi ketika pengajar sedang mengajar dan murid sedang belajar. Tipe ini lebih sering digunakan saat seminar atau konferensi atau yang lebih dikenal dengan kuliah *online*.

2. *Asynchronous training*

Asynchronous berarti “tidak pada waktu yang bersamaan”. Jadi seseorang dapat mengambil pelatihan pada waktu yang berbeda dengan pengajar memberikan pelatihan. Pelatihan ini lebih populer di dunia *mobile learning* karena memberikan keuntungan lebih bagi peserta, karena dapat mengakses pelatihan kapanpun dan dimanapun.

2.2.3 Konsep *Mobile Learning*

Efendi dan Zhuang (2005), mengusulkan sebuah konsep *mobile learning* pada sebuah institusi pendidikan sebagai berikut:

1. Konsep *mobile learning* difokuskan untuk menyediakan kelas pembelajaran maya yang memungkinkan interaksi antara guru dan siswa. Interaksi meliputi penyediaan materi ajar, ruang diskusi, penyampaian tugas dan pengumuman penilaian.
2. Teknologi yang diadopsi sebaiknya efektif secara manajemen pengajaran dan dinilai sebagai sebuah pembaharuan. Selain itu teknologi yang dipilih sebaiknya mudah di akses dan tersedia dengan distribusi yang merata di lingkungan siswa maupun guru.

2.2.4 Keuntungan *Mobile Learning*

Menurut Efendi dan Zhuang (2005), ada beberapa keuntungan dari penggunaan *mobile learning*, diantaranya sebagai berikut:

1. Biaya
Penghematan biaya bisa dilakukan karena dapat menekan biaya untuk urusan teknis yang biasa digunakan seperti pembelajaran konvensional, seperti penyediaan peralatan tulis, papan, konsumsi untuk pengajar, proyektor, dan lainnya.
2. Fleksibilitas waktu dan tempat
Mobile learning dapat membuat penggunanya menyesuaikan waktu dan tempat belajar. Mereka bisa menyisipkan pembelajaran disaat waktu luang dan tempat yang berbeda.

3. Standarisasi pengajaran

Adanya perbedaan kemampuan dalam memberikan pengajaran oleh guru atau pengajar menyebabkan peserta memiliki perbedaan dalam menyerap pembelajaran, terkadang standar pengajaran juga tergantung suasana hati pengajar. *Mobile learning* dapat menghapus perbedaan tersebut, pelajaran di *mobile learning* memiliki kualitas yang sama setiap kali diakses dan tidak tergantung suasana hati pengajar.

4. Fleksibilitas kecepatan pembelajaran

Setiap pelajar memiliki kemampuan yang berbeda dalam menyerap pelajaran, ada yang cepat dan ada yang lambat. Hal ini bisa diatasi oleh *mobile learning*, karena kecepatan belajar tergantung dari masing-masing pelajar.

2.2.5 Keterbatasan Mobile Learning

Walaupun *mobile learning* menawarkan beberapa keuntungan bagi penggunanya. Praktik ini juga memiliki beberapa keterbatasan yang harus diwaspadai oleh pengelola *mobile learning*, berikut beberapa keterbatasan dalam *mobile learning* (Efendi dan Zhuang, 2005):

1. Budaya

Sebagian orang merasa tidak nyaman mengikuti sistem *mobile learning*. Penggunaan *mobile learning* menuntut budaya *self-learning*, dimana seseorang memotivasi diri sendiri agar mau belajar. Sebaliknya, pada sebagian besar budaya pendidikan Indonesia, motivasi belajar banyak tergantung dari pengajar. Disini kita juga harus melihat kebiasaan penggunaan teknologi dari pelajar. Apabila mereka tidak terbiasa menggunakan perangkat mobile, maka penerapan *mobile learning* akan memakan waktu yang lama.

2. Investasi

Meskipun *mobile learning* menghemat banyak biaya, tapi suatu institusi harus mengeluarkan investasi awal yang cukup besar untuk memulai penerapan *mobile learning*. Investasi dapat berupa biaya perancangan dan pembuatan program *learnig management system*, paket pembelajaran, promosi dan lainnya.

3. Teknologi

Karena teknologi yang digunakan beragam ada kemungkinan teknologi tersebut tidak sejalan dengan yang sudah ada dan terjadi konflik teknologi sehingga mobile learning tidak berjalan dengan baik.

4. Infrastruktur

Mobile learning tidak lepas dari segi konektivitas, salah satunya akses internet yang berguna untuk mengakses konten yang ada di server. Belum meratanya layanan koneksi internet menjadi sebuah keterbatasan dalam penerapan *mobile learning*.

5. Materi

Tidak semua materi bisa dimasukkan dalam *mobile learning*, beberapa materi memerlukan adanya kegiatan fisik, seperti olahraga, instrumen musik, seni rupa, tari dan lainnya

2.3 Moodle

Moodle (Modular Object-Oriented Dynamic Learning Environment) merupakan paket perangkat lunak yang dibangun untuk kegiatan belajar berbasis internet dan website. *Moodle* merupakan salah satu aplikasi dari konsep dan mekanisme belajar mengajar yang memanfaatkan teknologi informasi, yang dikenal dengan konsep *e-Learning*. *Moodle* dapat digunakan secara bebas sebagai produk sumber terbuka *open source* di bawah lisensi GNU. (<http://Moodle.org>)

Dengan konsep yang ditawarkan oleh *Moodle*, Seorang guru/pengajar dapat memberikan materi dari mana saja. Begitu juga seorang siswa dapat mengikuti pelajaran dari mana saja. Dengan fasilitas yang tersedia pada *Moodle* seorang pengajar bisa melengkapi materi bahan ajarnya dengan menambahkan video, animasi, *power point*, forum diskusi *online*, *chatting* dan lain-lain yang memungkinkan pembelajaran bisa dilaksanakan secara menarik dan interaktif.

Moodle pertama kali dikembangkan oleh oleh Martin Dougiamas pada Agustus 2002 dengan *Moodle* Versi 1.0. *Moodle* dapat diakses dan di download di <http://Moodle.org>. *Moodle* dibangun berdasarkan filosofi *Social Constructionist*

Pedagogy yang artinya cara belajar yang baik adalah dari sudut pandang pelajar sendiri.

Moodle memiliki beberapa keunggulan yang membuat *Learning Management system* ini banyak digunakan oleh institusi pendidikan. Berikut beberapa keunggulan *Moodle*:

1. Sederhana, efisien, ringan, dan banyak sesuai dengan berbagai .
2. Mudah cara instalasinya dan banyak mendukung bahasa, salah satunya bahasa Indonesia.
3. Tersedianya manajemen situs untuk pengaturan situs keseluruhan.
4. Tersedianya manajemen pengguna.
5. Manajemen kursus, penambahan jenis kursus, pengurangan atau pengubahan.
6. Adanya modul chat, modul forum, modul jurnal, survai, workshop dan lainnya.
7. *Free dan Open Source*

2.3.1 Fitur Moodle

Sebagai sebuah perangkat lunak untuk pembelajaran dengan sistem *e-Learning*, *Moodle* memiliki fitur yang bagus dalam mendukung kegiatan belajar dengan sistem *e-Learning*. Berikut beberapa fitur *Moodle* (<http://Moodle.org>) :

1. Course.
2. Assignment submissions.
3. Upload file.
4. Chat.
5. Glossary.
6. Forum diskusi.
7. Kuis online.
8. Berita.
9. Wiki.
10. Multimedia.
11. Kalender online.

2.3.2 Karakteristik Pengguna Moodle

Dalam *Moodle*, pengguna terbagi dalam beberapa karakteristik. Setiap karakteristik memiliki hak akses masing-masing terhadap konten yang ada di *Moodle*. Berikut karakteristik pengguna *Moodle* (<http://Moodle.org>):

1. *Administrator*

Administrator memiliki hak penuh dalam mengakses semua konten *Moodle*. Biasanya bisa melakukan apapun di dalam sistem *Moodle*.

2. *Course creator*

Course creator memiliki hak akses sebatas membuat *course*. Tugas *user* ini membuat sebuah *course*.

3. *Teacher*

Teacher memiliki hak akses terhadap *course* yang di ajarkannya, termasuk memberi tugas, kuis, dan nilai.

4. *Non-editing teacher*

User kategori ini hanya memiliki akses dalam *course* dan menilai *student*, tapi tidak bisa untuk aktivitas memberi tugas, kuis atau lainnya.

5. *Student*

Student memiliki hak akses terhadap *course* yang telah di *enrol*.

6. *Guest*

User kategori ini hanya memiliki hak untuk melihat konten dari *Moodle* saja, dan tidak bisa melakukan kegiatan *input* data maupun teks.

7. *Authenticated user*

Semua *user* yang telah teridentifikasi melakukan akses terhadap sistem *Moodle*.

2.4 Konvergensi Perangkat Mobile

Konvergensi secara harfiah dapat diartikan dua benda atau lebih bertemu pada satu titik. Secara umum konvergensi adalah penyatuan berbagai layanan teknologi komunikasi serta informasi dalam satu perangkat.

Teknologi informasi mutakhir telah berhasil menggabungkan sifat-sifat teknologi telekomunikasi konvensional yang bersifat massif dengan teknologi komputer yang bersifat interaktif. Fenomena ini disebut sebagai konvergensi.

Konvergensi dapat merujuk pada teknologi yang sebelumnya terpisah, seperti suara (dan fitur telepon), data (dan produktivitas aplikasi) dan video, yang sekarang berbagi sumber daya dan berinteraksi antara satu dengan yang lain, sinergis menciptakan efisiensi baru.

Kunci dari konvergensi adalah digitalisasi, karena seluruh bentuk informasi maupun data diubah dari format analog ke format digital sehingga dikirim ke dalam satuan bit (binary digit). Karena informasi yang dikirim merupakan format digital, konvergensi mengarah pada penciptaan produk-produk yang aplikatif yang mampu melakukan fungsi audiovisual sekaligus. Maka saat ini banyak ditemukan perangkat komputer dapat difungsikan sebagai pesawat televisi, atau telepon genggam dapat menerima suara, tulisan, data maupun gambar tiga dimensi (3G).

2.5 *Android*

Android merupakan sebuah sistem operasi untuk perangkat *mobile* berbasis *linux* yang meliputi sistem operasi, *middleware*, dan aplikasi yang dirilis oleh Google. Sedangkan *Android SDK (Software Development kit)* menyediakan Tools dan API (*Application Programming Interface*) yang diperlukan untuk mengembangkan aplikasi pada *platform Android* dengan menggunakan bahasa pemrograman Java (Mulyadi, 2010). *Android* dikembangkan oleh Google bersama OHA (*Open Handset Alliance*) yaitu aliansi perangkat selular terbuka yang terdiri dari 47 perusahaan *Hardware*, *Software* dan perusahaan telekomunikasi ditujukan untuk mengembangkan standar terbuka bagi perangkat selular.

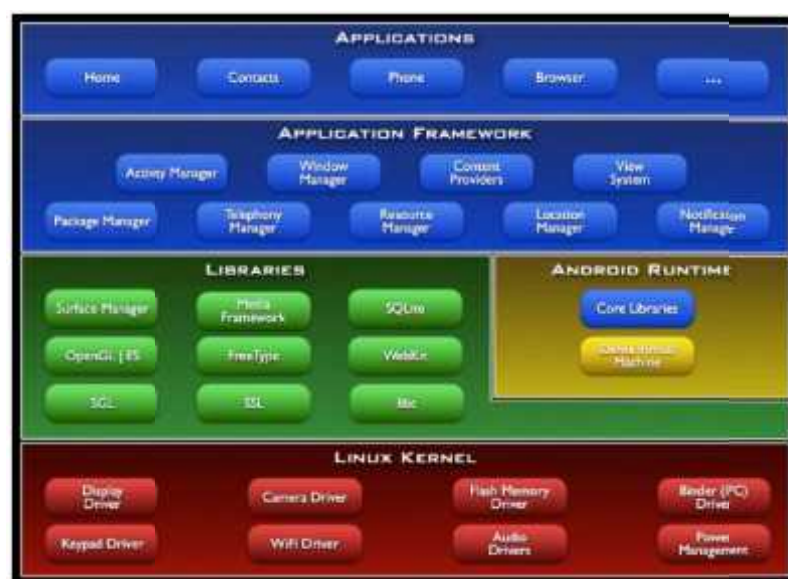
2.5.1 Sejarah *Android*

Pada tahun 2005 Google mengakuisisi *Android Inc* yang pada saat itu dimotori oleh Andy Rubin, Rich Miner, Nick Sears dan Chris White. Kemudian

pada tahun itu juga memulai membangun platform *Android* secara intensif. Kemudian pada tanggal 5 November 2007 *Android* pertama kali diluncurkan, dan *smartphone* pertama yang menggunakan sistem operasi *Android* dikeluarkan oleh T-Mobile dengan sebutan G1 pada bulan September 2008. Hingga saat ini *Android* telah merilis beberapa versi *Android* untuk menyempurnakan versi sebelumnya. Selain berdasarkan penomoran, pada setiap versi *Android* terdapat kode nama berdasarkan nama-nama kue. Hingga saat ini sudah terdapat beberapa versi yang telah diluncurkan, diantaranya: versi 1.5 dirilis pada 30 April 2009 diberi nama Cupcake, versi 1.6 dirilis pada 15 September 2009 diberi nama Donut, versi 2.0/2.1 dirilis pada 26 Oktober 2009 diberi nama Éclair, versi 2.2 dirilis pada bulan Mei 2010 diberi nama Froyo dan versi 2.3 dirilis pada Desember 2010 yang diberi nama Gingerbread.

2.5.2 Arsitektur *Android*

Dalam paket sistem operasi *Android* terdiri dari beberapa unsur seperti tampak pada gambar 2.1. Secara sederhana arsitektur *Android* merupakan sebuah kernel Linux dan sekumpulan pustaka C / C++ dalam suatu *framework* yang menyediakan dan mengatur alur proses aplikasi.



Gambar 2.1 Arsitektur *Android* (sumber: <http://developer.android.com>)

2.5.2.1 *Linux Kernel*

Android bukan *linux*, akan tetapi *android* dibangun diatas *linux kernel* versi 2.6 yang keandalannya sudah teruji. Untuk inti sistem servis *linux* yang digunakan seperti keamanan, manajemen momori, proses manajemen, *network* dan *driver* model. Seperti yang terlihat di gambar 2.6, *linux kernel* menyediakan *Driver* layar, *driver* kamera, flash memori, IPC (*Interprocces Communication*) untuk mengatur aplikasi dan keamanan, *driver* keypad, *driver* WiFi, *driver* audio, dan power management. *Kernel* juga bertindak sebagai lapisan abstrak antara *hardware* dan *software* stacknya (<http://deveoper.android.com>).

2.5.2.2 *Libraries*

Android menyertakan linbraries C / C++ yang digunakan oleh berbagai komponen dari sistem *Android*. Kemampuan ini disediakan kepada *developer* aplikasi melalui *framework* aplikasi *Android*. Beberapa inti libraries tercantun dibawah ini (Mulyadi, 2010):

1. *System C library*

Variasi dari implementasi BSD (*Barkeley Software Distribution*) berasal dari sistem standar C library (*libc*), sesuai untuk perangkat *embedded* berbasis *linux*.

2. *Media libraries*

Untuk merekam dan memutar berbagai format *Audio* dan *Video*.

3. *Surface Manager*

Mengelola akses ke subsistem layar, lapisan komposit 2D dan grafis 3D dari beberapa aplikasi.

4. *LibWebCore*

Librari untuk mesin web pada browser *Android*.

5. *SGL*

Merupakan library untuk proses mesin grafis 2D.

6. *3D libraries*

Digunakan untuk proses gambar yang membutuhkan daya 3D.

7. *FreeType*

Merupakan library untuk bitmap dan vektor font rendering

8. *SQLite*

Merupakan library mesin database yang kuat dan ringan, dan pehubung aplikasi yang tersedia.

2.5.2.3 *Android Runtime*

Pada *Android* tertanam paket pustaka inti yang menyediakan sebagian besar fungsi *Android*. Inilah yang membedakan *Android* dibandingkan dengan sistem operasi lain yang juga mengimplementasikan Linux. *Android Runtime* merupakan mesin virtual yang membuat aplikasi *Android* menjadi lebih tangguh dengan paket pustaka yang telah ada. Dalam *Android Runtime* terdapat 2 bagian utama, diantaranya (<http://developer.android.com>):

1. Pustaka Inti, *Android* dikembangkan melalui bahasa pemrograman Java, tapi *Android Runtime* bukanlah mesin virtual Java. Pustaka inti *Android* menyediakan hampir semua fungsi yang terdapat pada pustaka Java serta beberapa pustaka khusus *Android*.
2. Mesin Virtual Dalvik, Dalvik merupakan sebuah mesin virtual yang dikembangkan oleh Dan Bornstein yang terinspirasi dari nama sebuah perkampungan yang berada di Iceland. Dalvik hanyalah interpreter mesin virtual yang mengeksekusi file dalam format *Dalvik Executable* (*.dex). Dengan format ini Dalvik akan mengoptimalkan efisiensi penyimpanan dan pengalamatan memori pada file yang dieksekusi. Dalvik berjalan diatas kernel Linux 2.6, dengan fungsi dasar seperti *threading* dan manajemen memori yang terbatas.

2.5.2.4 *Applications Frameworks*

Android merupakan “*Open Development Platform*” dimana *Android* menawarkan kepada pengembang untuk membangun aplikasi yang bagus dan inovatif. Pengembang bebas untuk mengakses perangkat keras, akses informasi resources, menjalankan service background, mengatur alarm, menambah status

pemberitahuan dan lainnya. Pengembang memiliki akses penuh menuju API framework seperti yang dilakukan oleh aplikasi yang kategori inti.

Applications Frameworks merupakan layer dimana para pembuat aplikasi melakukan pengembangan aplikasi yang akan dijalankan di sistem operasi *android*. Komponen-komponen yang termasuk dalam *Applications Frameworks* adalah sebagai berikut (<http://developer.android.com>):

1. *Views*
2. *Content Provider*
3. *Resource Manager*
4. *Notification Manager*
5. *Activity Manager*

2.5.2.5 Applications dan Widgets

Applications dan *Widgets* ini adalah layer yang berhubungan dengan aplikasi saja. Biasanya aplikasi di download kemudian dilakukan instalasi dan jalankan aplikasi tersebut. Di layer ini terdapat aplikasi inti termasuk klien email, program SMS, kalender, peta, browser, kontak, dan lain-lain.

2.5.3 Android SDK (Software Development Kit)

Android SDK adalah tools API (*Application Programming Interface*) yang diperlukan untuk memulai pengembangan aplikasi pada *platform Android* menggunakan bahasa pemrograman Java. *Android* merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, *middleware* dan aplikasi kunci yang dirilis oleh Google. Untuk sumber SDK *Android* dapat dilihat dan diunduh langsung ke situs resmi *Android* di <http://developer.android.com>.

2.5.4 Komponen Aplikasi

Aplikasi *Android* ditulis dalam bahasa pemrograman Java. Kode Java dikompilasi bersama dengan data file resource yang dibutuhkan oleh aplikasi, prosesnya di paket oleh *tools* yang dinamakan *atp tools* kedalam paket *Android*,

sehingga menghasilkan file dengan ekstensi *apk*. Ada 4 jenis komponen pada aplikasi *Android*, yaitu (<http://deveoper.android.com>):

1. *Activities*

Suatu *Activity* akan menyajikan *user interface* (UI) kepada pengguna, sehingga pengguna dapat melakukan interaksi. Pada umumnya sebuah aplikasi *Android* memiliki banyak *activity*, tergantung pada tujuan aplikasi dan desain dari aplikasi tersebut. Satu *activity* biasanya akan dipakai untuk menampilkan aplikasi atau yang bertindak sebagai *user interface* (UI) saat aplikasi diperlihatkan kepada *user*. Untuk pindah dari satu *activity* ke *activity* lain, kita dapat melakukan dengan satu even, misalnya klik tombol, memilih opsi atau menggunakan triggers tertentu. Secara hirarki sebuah windows *activity* dinyatakan dengan *method* `Activity setContentView()`. `ContentView` adalah objek yang berada pada *root hirarki*.

2. *Service*

Service tidak memiliki *Graphic User Interface* (GUI), tetapi *service* berjalan secara background. Komponen *service* diproses tidak terlihat, memperbaharui sumber data dan menampilkan notifikasi. *Service* digunakan untuk melakukan pengolahan data yang terus diproses, bahkan ketika *activiti* tidak aktif

3. *Broadcast Receiver*

Broadcasr receiver berfungsi menerima dan bereaksi untuk menyampaikan notifikasi. *Broadcast receiver* tidak memiliki *user interface*, tetapi memiliki sebuah *activity* untuk merespon informasi yang diterima atau menggunakan *notification manager* untuk memberi tahu pengguna, seperti lampu latar atau getaran.

4. *Content Provider*

Content provider membuat kumpulan aplikasi data secara spesifik sehingga bisa digunakan oleh aplikasi lain. Data disimpan dalam file seperti database *SQLite*. *Content provider* menyediakan cara untuk mengakses data yang dibutuhkan oleh suatu *activity*, misalnya ketika kita menggunakan aplikasi

yang membutuhkan peta atau aplikasi yang membutuhkan untuk mengakses data kontak dan navigasi, maka disinilah fungsi dari content provider.

2.5.5 Kelebihan *Platform Android*

Persaingan *platform* atau sistem operasi semakin ketat, ini dapat dilihat dari banyaknya sistem operasi yang ada seperti, Symbian, iPhone, Windows Mobile, BlackBerry, Java Mobile Edition, Linux Mobile (LiM0), dan banyak lagi. Namun ada beberapa hal yang menjadi kelebihan *Android*. Walaupun beberapa fitur-fitur yang ada telah muncul sebelumnya pada platform lain, *Android* adalah yang pertama menggabungkan hal seperti berikut (Amiral, 2010) :

1. Keterbukaan, Bebas pengembangan tanpa dikenakan biaya terhadap sistem karena berbasiskan Linux dan *open source*. Pembuat perangkat menyukai hal ini karena dapat membangun *platform* yang sesuai yang diinginkan tanpa harus membayar *royalty*. Sementara pengembang *software* menyukai karena *Android* dapat digunakan diperangkat manapun dantampa terikat oleh vendor manapun.
2. Arsitektur komponen dasar *Android* terinspirasi dari teknologi internet *Mashup*. Bagian dalam sebuah aplikasi dapat digunakan oleh aplikasi lainnya, bahkan dapat diganti dengan komponen lain yang sesuai dengan aplikasi yang dikembangkan.
3. Banyak dukungan *service*, kemudahan dalam menggunakan berbagai macam layanan pada aplikasi seperti penggunaan layanan pencarian lokasi, database SQL, browser dan penggunaan peta. Semua itu sudah tertanam pada *Android* sehingga memudahkan dalam pengembangan aplikasi.
4. Siklus hidup aplikasi diatur secara otomatis, setiap program terjaga antara satu sama lain oleh berbagai lapisan keamanan, sehingga kerja sistem menjadi lebih stabil. Pengguna tak perlu khawatir dalam menggunakan aplikasi pada perangkat yang memorinya terbatas.
5. Dukungan grafis dan suarat terbaik, dengan adanya dukungan 2D grafis dan animasi yang diilhami oleh *Flash* menyatu dalam 3D menggunakan *OpenGL* memungkinkan membuat aplikasi maupun game yang berbeda.

6. Portabilitas aplikasi, aplikasi dapat digunakan pada perangkat yang ada saat ini maupun yang akan datang. Semua program ditulis dengan menggunakan bahas pemrograman Java dan dieksekusi oleh mesin virtual Dalvik, sehingga kode program portabel antara ARM, X86, dan arsitektur lainnya. Sama halnya dengan dukungan masukan seperti penggunaan *Keyboard*, layar sentuh, *trackball* dan resolusi layar semua dapat disesuaikan dengan program.

2.6 Analisa dan Perancangan Berorientasi Objek

Teknologi objek menganalogikan sistem aplikasi seperti kehidupan nyata yang didominasi oleh objek. Didalam membangun sistem berorientasi objek akan menjadi lebih baik apabila langkah awalnya didahului dengan proses analisis dan perancangan yang berorientasi objek. Tujuannya adalah untuk mempermudah *programmer* didalam mendesain program dalam bentuk objek-objek dan hubungan antar objek tersebut untuk kemudian dimodelkan dalam sistem nyata (A.Suhendar, 2002).

perusahaan *software*, Rational Software, telah membentuk konsorsium dengan berbagai organisasi untuk meresmikan pemakaian *Unified Modelling Language* (UML) sebagai bahasa standar dalam *Object Oriented Analysis Design* (OOAD).

2.6.1 *Unified Modelling Language* (UML)

Unified Modelling Language (UML) adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem (Dharwiyanti dan Wahono, 2006).

Untuk merancang sebuah model, UML memiliki beberapa diagram antara lain : *use case diagram*, *class diagram*, *statechart diagram*, *activity diagram*, *sequence diagram*, *collaboration diagram*, *component diagram*, *deployment diagram*.

2.6.1.1 *Use Case Diagram*

Use case diagram merupakan sebuah gambaran fungsionalitas sebuah sistem. Sebuah use case merepresentasikan interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, *create* sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu Dharwiyanti (2006).

Dalam sebuah sistem *use case diagram* akan sangat membantu dalam hal menyusun *requirement*, mengkomunikasikan rancangan dengan klien dan merancang *test case* untuk semua fitur yang ada pada sistem.

2.6.1.2 *Class Diagram*

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi) Dharwiyanti (2006).

Class diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class* memiliki tiga area pokok yaitu, nama, *stereotype*, atribut dan metoda

2.6.1.3 *Sequence Diagram*

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang memicu aktivitas tersebut, proses dan perubahan

apa saja yang terjadi secara internal dan *output* apa yang dihasilkan Dharwiyanti (2006).

2.6.1.4 Deployment Diagram

Deployment/physical diagram menggambarkan detail bagaimana komponen di-*deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi *server*, dan hal-hal lain yang bersifat fisikal Sebuah *node* adalah server, *workstation*, atau piranti keras lain yang digunakan untuk men-*deploy* komponen dalam lingkungan sebenarnya. Hubungan antar *node* (misalnya TCP/IP) dan *requirement* dapat juga didefinisikan dalam diagram ini Dharwiyanti (2006).

2.6.1.5 Activity Diagram

Activity diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi Dharwiyanti (2006).

Activity diagram merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

2.7 Rational Unified Process (RUP)

Untuk pengembangan aplikasi *mobile learning client server* berbasis Moodle pada Platform Android pada tugas akhir ini menggunakan metode pengembangan perangkat lunak Rational Unified Process (RUP).

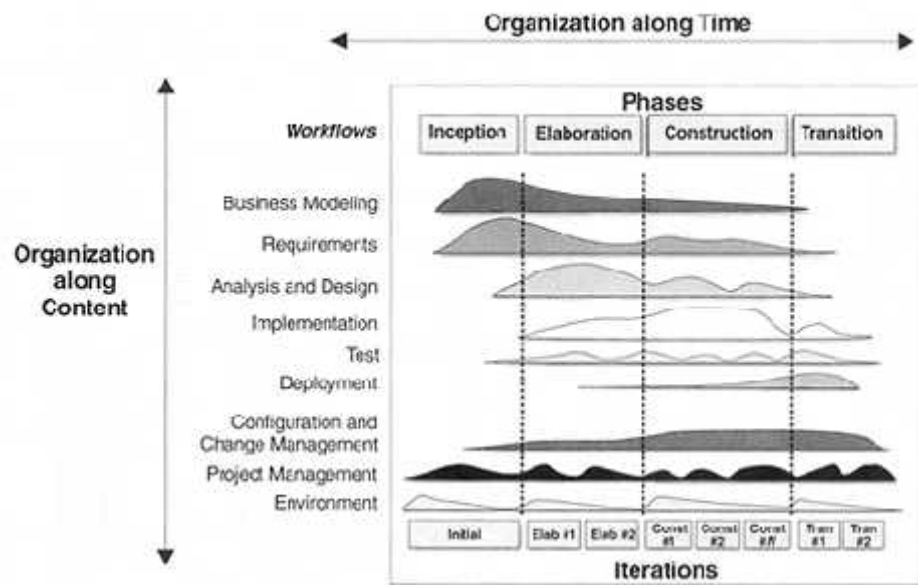
2.7.1 Pengertian RUP

Rational Unified Process adalah sebuah Proses Rekayasa Perangkat Lunak. RUP menyediakan pendekatan disiplin untuk memberikan tugas dan tanggung jawab dalam organisasi pengembang perangkat lunak. Tujuannya untuk memastikan perangkat lunak yang berkualitas tinggi dan sesuai kebutuhan penggunaanya dalam anggaran dan jadwal yang dapat diprediksi (Kruchten, 2000).

RUP mengarahkan kita terhadap pengembang perangkat lunak secara praktis dan efektif. Terdapat 6 *best practice* atau disebut juga *basic principle* dalam metode RUP, antara lain (Kruchten, 2000):

1. *Develop software iteratively*, bertujuan untuk mengurangi resiko pada awal proyek.
2. *Manage requirements*, bertujuan untuk mengatur kebutuhan yang diperlukan selama proyek.
3. *Use component-based architectures* untuk membangun komponen arsitektur sebuah proyek.
4. *Visually model software*, bertujuan untuk merancang sebuah model visual perangkat lunak, untuk mendapatkan struktur dan perilaku dari aritektur perangkat lunak.
5. *Continuously verify software quality*,
6. *Control changes to software*. kemampuan untuk mengatur serta mengubah perangkat lunak saat dibutuhkan.

RUP menggunakan konsep *object oriented*, dengan aktifitas yang berfokus pada pengembangan model dengan menggunakan *Unified Model Language* (UML). Melalui gambar 2.2 dibawah dapat dilihat bahwa RUP memiliki 2 dimensi, yaitu:



Gambar 2.2 Struktur proses 2 dimensi RUP (*sumber: Kruchten, 2000*)

Dimensi pertama digambarkan secara horizontal. Dimensi ini mewakili aspek-aspek dinamis dari pengembangan perangkat lunak. Aspek ini dijabarkan dalam tahapan pengembangan atau fase. Setiap fase akan memiliki suatu *major milestone* yang menandakan akhir dari awal dari phase selanjutnya. Setiap phase dapat berdiri dari satu atau beberapa iterasi. Dimensi ini terdiri atas *Inception*, *Elaboration*, *Construction*, dan *Transition*.

Dimensi kedua digambarkan secara vertikal. Dimensi ini mewakili aspek-aspek statis dari proses pengembangan perangkat lunak yang dikelompokkan ke dalam beberapa disiplin. Proses pengembangan perangkat lunak yang dijelaskan kedalam beberapa disiplin terdiri dari empat elemen penting, yakni *who is doing*, *what*, *how* dan *when*. Dimensi ini terdiri atas *Business Modeling*, *Requirement*, *Analysis and Design*, *Implementation*, *Test*, *Deployment*, *Configuration* dan *Change Manegement*, *Project Management*, *Environtment*.

2.7.2 Fase RUP

Fase-fase pada RUP berdasarkan waktu pengerjaan proyek dapat dibagi menjadi 4 fase, yaitu *Inception*, *Elaboration*, *Construction* dan *Transition* (Rational Team, 2001).

1. Fase *Inception*

Fase *inception* merupakan fase untuk mengidentifikasi masalah, untuk itu diperlukan juga idetifikasi entitas dari luar yang berhubungan dengan sistem. Pada fase ini melibatkan semua identifikasi *use case* dan gambaranya. Selain itu juga termasuk kriteria keberhasilan proyek, perkiraan resiko, perkiraan terhadap *resource* yang dibutuhkan dan merencanakan penjadwalan *milestone*. Hasil yang diperoleh pada fase ini adalah :

- a. Dokumen visi (visi dari kebutuhan proyek, kata kunci, batasan utama).
- b. Inisialisasi model *use-case* (10%-20% selesai).
- c. Daftar kata.
- d. *Business case*, termasuk didalamnya konteks bisnis, kriteria sukses, pengenalan pasar dan proyeksi keuangan.
- e. Inisialisasi penialaian resiko.
- f. Rencana proyek dan menunjukan fase serta iterasi.
- g. Model bisnis jika diperlukan

Kriteria evaluasi untuk fase *Inception* adalah :

- a. Menyesuaikan *stakeholder* dengan *scope definition* dan perkiraan biaya atau perkiraan jadwal.
- b. Pemahaman terhadap *use-case* utama
- c. Kredibilitas dari perkiraan biaya, jadwal, prioritas, resiko dan proses pengembangan.
- d. Pemahaman terhadap *prototype*

2. Fase *Elaboration*

Tujuan dari fase *elaboration* (pengembangan) adalah menganalisa area permasalahan, mengembangkan rencana proyek, dan menghilangkan unsur-unsur yang memiliki resiko besar terhadap proyek. Adapun hasil dari fase *elaboration* adalah:

- a. *Use case* model, seluruh use case dan aktor telah teridentifikasi.
- b. *Requirement* tambahan yang mungkin tidak bersifat fungsional bagi proyek.
- c. *Software Architecture Description* (Deskripsi Arsitektur Perangkat Lunak).
- d. Prototipe dari arsitektur yang dapat dieksekusi.
- e. Revisi daftar tingkat resiko dan revisi *business-case*.
- f. Rencana pengembangan keseluruhan proyek.
- g. Persiapan dokumen panduan bagi pengguna (*user manual*).

Kriteria utama dalam fase elaboration melibatkan pertanyaan berikut :

- a. Apakah produk sudah stabil ?
- b. Apakah rancangan arsitekturalnya sudah stabil ?
- c. Apakah saat demo prototipe, unsur yang memiliki resiko telah bisa di atur ?
- d. Apakah rencana konstruksi telah detail dan akurat ?
- e. Apakah *stakeholder* bersedia dan menyepakati visi dari pengembangan proyek tersebut?
- f. Apakah pembelanjaan *actual-resource* terhadap rencana pembelanjaan dapat diterima?

3. Fase *Contruction*

Selama fase konstruksi, semua komponen dan fitur yang dikembangkan terintegrasi ke dalam produk dan secara menyeluruh semua fitur telah diuji. Di lain sisi, proses konstruksi adalah sebuah proses *manufacturing*, dimana terdapat penekanan dalam mengelola *resource* dan mengatur operasi untuk mengoptimalkan jadwal dan kualitas. Pada tahap ini pola pikir (*mindset*) mengalami perubahan dari pengembangan *intellectual property* pada fase *Inception* dan *Elaboration*, menjadi pengembangan *deployoable product*. Kriteria evaluasi terhadap fase *Construction* ini adalah :

- a. Apakah peluncuran produk cukup baik dan dapat diterima di komunitas pengguna?

- b. Apakah semua *stakeholder* siap untuk beralih ke komunitas pengguna?
- c. Apakah pembelanjaan *actual-resource* terhadap rencana pembelanjaan masih tetap diterima?

4. Fase *Transition*

Tujuan dari fase ini adalah untuk transisi dari produk perangkat lunak ke pengguna akhir. Apabila produk telah di luncurkan kepada pengguna, maka isu-isu akan muncul dari pengguna. Nantinya isu ini akan digunakan untuk tahap perbaikan terhadap produk. Kriteria evaluasi untuk fase *Transition* adalah :

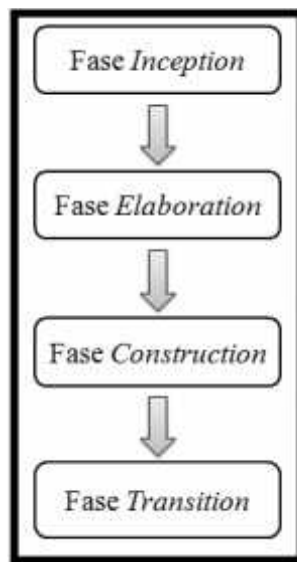
- a. Apakah pengguna merasa puas?
- b. Apakah pembelanjaan *actual-resource* terhadap rencana pembelanjaan masih tetap diterima?

BAB III

METODOLOGI PENELITIAN

3.1. Tahapan Penelitian

Tahapan penelitian yang akan dilaksanakan pada Tugas Akhir ini menggunakan metode *Rational Unified Process* (RUP), seperti yang telah dijelaskan pada bab landasan teori bahwa RUP merupakan suatu metode yang digunakan untuk proses pembangunan sebuah perangkat lunak. Gambar 3.1 di bawah ini menjelaskan tahapan penelitian terhadap sistem *mobile learning client server* berbasis *moodle* pada *platform android* yang akan dibangun berdasarkan kepada metode RUP.



Gambar 3.1. Tahapan penelitian dengan metode RUP.

3.2. Tahapan *Rational Unified Process* (RUP)

Berikut ini akan diuraikan tahapan-tahapan pembuatan aplikasi *mobile learning client server* berbasis *Moodle* pada *Platform Android* menggunakan metode *Rational Unified Process* (RUP).

3.2.1. Fase *Inception*

Fase *inception* merupakan fase untuk mengidentifikasi masalah yang ada pada sistem *mobile learning client server* berbasis *moodle* pada *platform android* yang akan bangun, untuk itu diperlukan juga identifikasi entitas dari luar yang berhubungan dengan sistem. Pada fase ini melibatkan semua identifikasi *use case* dan gambaranya. Pada fase ini akan dilakukan tugas-tugas sebagai berikut:

- a. Pengenalan masalah, yaitu memahami permasalahan yang terjadi sistem *mobile learning client server* berbasis *moodle* pada *platform android* yang akan dibangun.
- b. Pembuatan proposal untuk pembangunan sistem *mobile learning client server* berbasis *moodle* pada *platform android*, yaitu mencakup latar belakang permasalahan, pokok permasalahan, tujuan dan batasan permasalahan.
- c. Studi literature yang berhubungan dengan pembangunan sistem, mencakup penelusuran teori-teori yang berhubungan dengan permasalahan dan teknik pembangunan sistem *mobile learning client server* berbasis *moodle* pada *platform android*.
- d. *Project Plan* untuk sistem *mobile learning* yang akan dibangun, yaitu mencakup jadwal pelaksanaan tugas-tugas yang akan dijalani.
- e. Penyelesaian model *use-case* (10%-20%) sistem.
- f. Contoh *prototype mobile learning* berbasis *moodle* pada *platform android*.

3.2.2. Fase *Elaboration*

Fase *elaboration* merupakan tahapan pengembangan dan analisa terhadap sistem *mobile learning* yang akan dibangun. Tujuan dari fase *elaboration* ini mengecilkan kemungkinan kesalahan yang akan terjadi. Pada fase *elaboration* akan dilakukan tugas-tugas sebagai berikut:

- a. Pembuatan deskripsi arsitektur aplikasi *mobile learning client server* berbasis *Moodle* pada *Platform Android*

- b. Deskripsi kebutuhan sistem *mobile learning client server* berbasis *moodle* pada *platform android*.
- c. Deskripsi pengguna.
- d. Penyempurnaan model *use-case* (> 80%).
- e. Penyempurnaan perancangan sistem dengan *Unified Modeling Language* (UML).
- f. Prototipe dari arsitektur yang dapat dieksekusi.

3.2.3. Fase *Construction*

Fase *Construction* merupakan tahapan pembangunan sistem mobile learning sesuai rencana yang telah dibuat pada tahap *elaboration*. Pada fase ini akan dilakukan tugas-tugas sebagai berikut:

- a. Membangun sistem *mobile learning client server* berbasis *moodle* pada *platform android* yang berpedoman pada model *use-case* menggunakan bahasa pemrograman Java.
- b. Implementasi sistem pada perangkat Android.
- c. Melakukan pengujian terhadap kesalahan-kesalahan yang mungkin akan terjadi selama proses membangun aplikasi.

3.2.4. Fase *Transition*

Fase *transition* merupakan tahap evaluasi dari sistem yang telah di rencanakan, dianalisa dan dibangun. Dari tahap ini bisa dilihat kekurangan dari sistem yang telah dibangun. Pada fase ini akan dilakukan tugas-tugas sebagai berikut:

- a. Melakukan pengujian aplikasi terhadap *server*, apakah telah bisa berkomunikasi baik dengan *server*.
- b. Melakukan pengujian dengan metode *black box*.
- c. Mengamati kekurangan yang ada terhadap aplikasi yang telah dilakukan pengujian.

- d. Menyimpulkan fitur-fitur tambahan guna pengembangan aplikasi *mobile learning client server* berbasis *Moodle* pada *Platform Android* untuk versi selanjutnya.

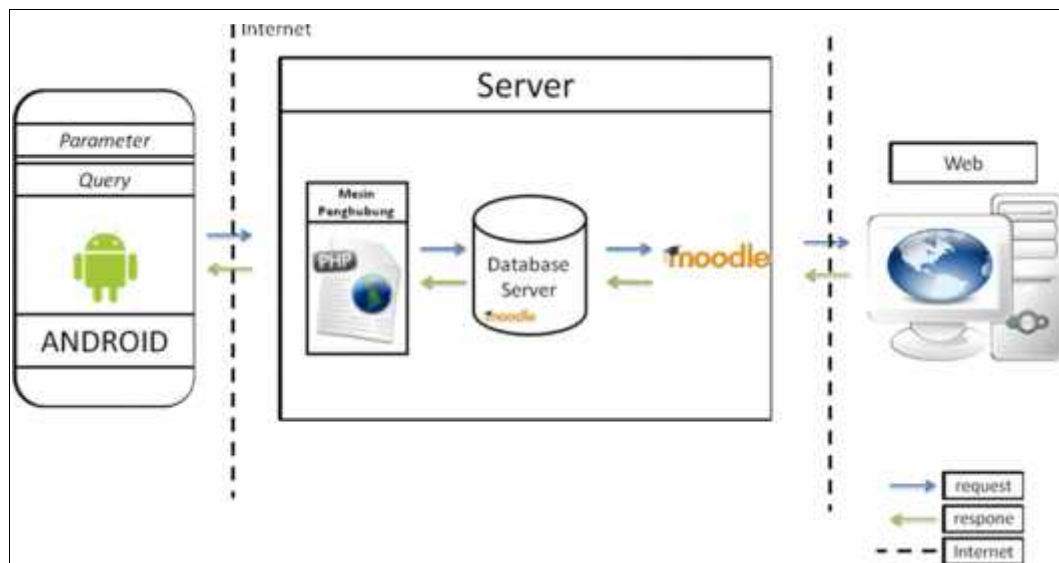
BAB IV

ANALISA DAN PERANCANGAN

4.1. Deskripsi Arsitektur Sistem

Arsitektur aplikasi *Mobile learning* disini menggambarkan sebuah sistem *mobile learning* yang berjalan pada perangkat yang terpasang sistem operasi *Android* seperti *smartphone*. Aplikasi ini menggunakan *Learning Management System Moodle* sebagai *server Mobile learning*, serta menggunakan bahasa pemrograman PHP sebagai mesin penghubung untuk mengakses *server*. Aplikasi ini berfungsi sebagai sistem yang mengakses dan memanipulasi isi *database Moodle*. Aplikasi yang terpasang di *smartphone Android* ini hanya bisa mengakses isi *course* berupa teks, *input dan update* tugas ke *database Moodle*, mengakses forum, mengakses file yang ada di *Moodle* untuk di unduh, dan mengakses nilai.

Deskripsi arsitektur ini bertujuan untuk memberikan gambaran mengenai modul-modul yang akan dirancang pada sistem ini dan hubungan antara modul-modul tersebut akan mempermudah proses perancangan sistem yang akan dibangun. Sistem dari sisi *client atau smartphone* yang akan dibangun adalah modul yang bisa mengirimkan *parameter* ke *server* serta modul yang berfungsi sebagai *parsing query* yang ke *server*. Untuk mesin penghubung antara *client* dan *server* berfungsi sebagai mesin yang bisa meneruskan parameter *query* dari *client* ke *server* dan sebaliknya. Untuk lebih jelasnya deskripsi arsitektur sistem ini dapat di lihat pada gambar 4.1. dibawah ini.



Gambar 4.1. Arsitektur sistem.

Dari gambar 4.1 diatas dapat dilihat proses kerja aplikasi yang akan dibuat, ada tiga bagian penting yang saling terhubung dalam kerja sistemnya, diantaranya:

1. *Device Android* merupakan perangkat tempat berjalannya aplikasi *Mobile learning*. Dari perangkat inilah pengguna berinteraksi dengan sistem, kemudian pengguna melakukan *input* parameter dan *query* dan sistem akan memproses parameter dan *query* tersebut untuk diteruskan ke *server* melalui PHP, dengan memanfaatkan jaringan GPRS untuk koneksinya.
2. *Server*, berfungsi sebagai mesin penghubung antara sistem yang berjalan di *Android* dengan *database server* Moodle. Peran dari mesin penghubung ini sangat vital, karena *Android* tidak bisa langsung menyentuh *database server* tanpa perantara. Kemudian PHP ini jugalah yang akan meneruskan *request* dari sistem di *Android* dan mengembalikan respon dari *database server* ke sistem di *Android*.
3. *Database server* merupakan bagian yang berfungsi sebagai *database* dari *mobile learning*. *Database server* ini yang bertanggung jawab memberikan konten *mobile learning* sesuai *request* dari sistem di *Android*. *Database*

yang digunakan merupakan *database* bawaan dari Moodle. Sehingga struktur dari *database* itu sendiri terbuat ketika kita memasang Moodle di *server*.

Sebelum mengakses sistem, pengguna harus terdaftar di *database server*. Untuk bisa terdaftar di *database*, pengguna harus melakukan pendaftaran melalui *web* setelah melakukan pendaftaran barulah si pengguna bisa masuk ke sistem di *Android*. Kemudian, untuk mendapatkan konten dari *mobile learning*, pengguna juga harus melakukan *enrol* manual terhadap *course* yang akan diambil melalui *web*. Setelah melakukan *enrol*, pengguna bisa mendapatkan konten dari *mobile learning*.

4.2. Deskripsi Kebutuhan Sistem

Untuk membangun sebuah sistem yang efisien, kebutuhan sistem merupakan hal yang harus diperhatikan. Mengetahui kebutuhan sistem akan membantu dalam pembangunan sistem.

4.2.1. Sistem yang akan dibangun

Untuk kebutuhan sistem yang akan dibangun terdapat dua bagian yakni dari kebutuhan sistem dari perangkat *Android* dan mesin penghubung menggunakan bahasa pemrograman PHP sebagai penghubung antara sistem di perangkat *Android* dengan *database server*. Dari sisi *server*, sistemnya telah terbentuk sendiri ketika memasang Moodle di *server*, sehingga bagian sisi *server* tidak dibahas dalam analisa sistem yang akan dibangun.

4.2.1.1. Sistem Pada Perangkat *Android*

Kebutuhan sistem pada perangkat *Android* adalah:

1. Bahasa pemrograman yang digunakan adalah *Java*.
2. Untuk masuk ke sistem, pengguna harus memasukan *username* dan *password*. *Username* dan *password* bisa didapatkan pengguna jika telah melakukan pendaftaran melalui *web*. Untuk itu dibutuhkan proses yang bisa menangani aksi masuk ke sistem nantinya.

3. Untuk menampilkan isi *course* yang diambil, pengguna cukup memilih *course* yang diambil, kemudian sistem di *Android* akan menampilkannya di perangkat *Android*. Isi *course* yang ditampilkan hanya mendukung teks. Untuk bisa mendapatkan *course* ini, pengguna harus melakukan enrol manual di *web*. Untuk itu dibutuhkan proses yang mampu menangani aksi menampilkan isi *course* berserta detailnya.
4. Untuk melihat tugas, pengguna cukup melakukan klik di *course* yang ada dan melihat apakah ada tugas. Jika ada maka pengguna akan melakukan *upload dan update* tugas baik dalam bentuk teks maupun berkas. Untuk itu diperlukan proses yang bisa menangani aksi *upload dan update* tugas tersebut.
5. Untuk *upload* tugas diberi pembatasan waktu *upload* tugas. Jika *student* melakukan *upload* melewati batas waktu yang ditentukan oleh *teacher*, maka tugas yang di *upload* tidak akan tersimpan di-*database*. Maka diperlukan proses yang bisa menangani aksi ini.
6. Untuk melihat nilai, pengguna cukup melakukan klik *course* dan melihat nilai berdasarkan tugas yang diberikan di masing-masing *course*. Maka diperlukan proses yang bisa menangani aksi untuk melihat nilai.
7. Untuk melakukan *download* file, pengguna cukup memilih *course* yang ada. Kemudian melihat file yang bisa di *download*. Untuk itu dibutuhkan proses yang bisa menangani aksi *download file*.
8. Untuk melihat berita, pengguna cukup memilih berita yang ada dan bisa melihat detail berita tersebut.

4.2.1.2. Mesin Penghubung

Mesin penghubung ini berfungsi sebagai perantara atau penghubung antara sistem yang berjalan di perangkat *Android* dengan *database server*. Mesin Penghubung ini dibangun menggunakan bahasa pemrograman PHP. Berikut beberapa Mesin yang akan dibangun:

1. Mesin *login*, berfungsi menangani *request login* dari sistem pada perangkat *Android* dan meneruskan ke *database server*. Kemudian

mengembalikan respon dari *database server* untuk diteruskan ke sistem di *Android*.

2. Mesin *course*, berfungsi menangani *request* menampilkan *course* beserta isi dari sistem di perangkat *Android* dan meneruskan ke *database server*. Kemudian mengembalikan respon dari *database server* untuk di kembalikan ke sistem di *Android*.
3. Mesin tugas, berfungsi menangani *request* menampilkan tugas dari sistem di perangkat *Android* dan meneruskan ke *database server*. Kemudian mengembalikan respon dari *database server* untuk diteruskan ke sistem di *Android*.
4. Mesin *upload* dan *update* tugas, berfungsi menangani *request upload* dari sistem di perangkat *Android* dan meneruskan ke *database server*. Kemudian mengembalikan ke sistem di *Android* dengan disertai pesan, file atau teks berhasil di *upload* atau tidak.
5. Mesin nilai, berfungsi menangani *request* menampilkan nilai dari sistem di perangkat *Android* dan meneruskan ke *database server* untuk di proses. Kemudian mesin akan menerima respon dan meneruskan respon ke sistem di *Android*.
6. Mesin *download* file, berfungsi menangani *request download* file dari sistem di perangkat *Android* dan meneruskan ke *database server*. Kemudian mengembalikan ke sistem di *Android*. Barulah si pengguna bisa melakukan *download* ketika mendapat respon dari *database server*.
7. Mesin berita, berfungsi menangani *request* menampilkan berita dari sistem di perangkat *Android*.

4.3. Fungsi Sistem

Secara umum fungsi sistem ada dua bagian yaitu sistem yang akan dibangun dari sisi perangkat *Android* dan mesin penghubung.

4.3.1. fungsi Sistem dari Sisi Perangkat *Android*

Sistem yang akan dibangun dari sisi perangkat *Android* memiliki fungsi-fungsi sebagai berikut:

1. Fungsi *login* ke sistem di perangkat *Android*.
2. Menampilkan *course* berserta isinya.
3. Menampilkan tugas berdasarkan *course* berserta isinya.
4. Menampilkan nilai berdasarkan tugas yang diberikan di setiap *course*
5. *Upload* dan *update* tugas berupa teks maupun file.
6. *Download* file berdasarkan *course* yang ada.
7. Menampilkan berita.

4.3.2. Fungsi Mesin Penghubung

Mesin Penghubung yang akan dibangun memiliki fungsi-fungsi sebagai berikut :

1. Fungsi untuk *login*.
2. Fungsi untuk memilih *course* yang akan ditampilkan.
3. Fungsi memilih tugas yang akan ditampilkan.
4. Fungsi *insert* tugas yang akan dikirim dari perangkat *Android*.
5. Fungsi memilih file untuk di *download*.
6. Fungsi menampilkan berita.

4.4. Deskripsi Pengguna

Pengguna dari sistem ini adalah *student* dan *teacher*. Tapi secara umum pengguna utamanya *student*, *student* hanya bisa mengakses isi dari konten *mobile learning* sesuai yang mereka enrol dan melakukan *input* berupa teks dan file. Untuk *teacher* sendiri sebagian kegiatannya banyak dilakukan di sisi *web*. Tapi tidak menghilangkan hak *teacher* untuk bisa mengakses *course* yang ada, karena *teacher* juga terdaftar dalam *database* user. Sehingga secara *default* bisa mengakses isi *mobile learning* dari perangkat *Android* dengan mendapatkan fitur yang sama dengan *student*. Untuk lebih jelasnya dapat di lihat di tabel 4.1. di bawah ini.

Tabel 4.1. Tabel deskripsi pengguna.

No	Kategori Pengguna	Hak Akses
1	<i>Student</i> (Perangkat <i>Android</i>)	a. <i>Login</i> ke sistem di perangkat <i>Android</i> . b. Melihat <i>course</i> serta isinya yang telah di <i>enrol</i> . c. Melihat tugas berdasarkan <i>course</i> . d. Melakukan <i>input</i> tugas berupa teks maupun <i>file</i> . e. Melihat nilai berdasarkan tugas pada masing-masing <i>course</i> . f. Melakukan <i>download file</i> . g. Melihat berita.
2	Teacher (Perangkat <i>Android</i>)	a. <i>Login</i> ke sistem di perangkat <i>Android</i> . b. Melihat <i>course</i> dan isinya, sesuai dengan <i>course</i> yang diajar oleh <i>teacher</i> . c. Melihat tugas, sesuai dengan <i>course</i> yang diajar oleh <i>teacher</i> . d. Melakukan <i>download file</i> , sesuai <i>course</i> yang diajar oleh <i>teacher</i> . e. Melihat berita.

4.5. Perancangan Sistem

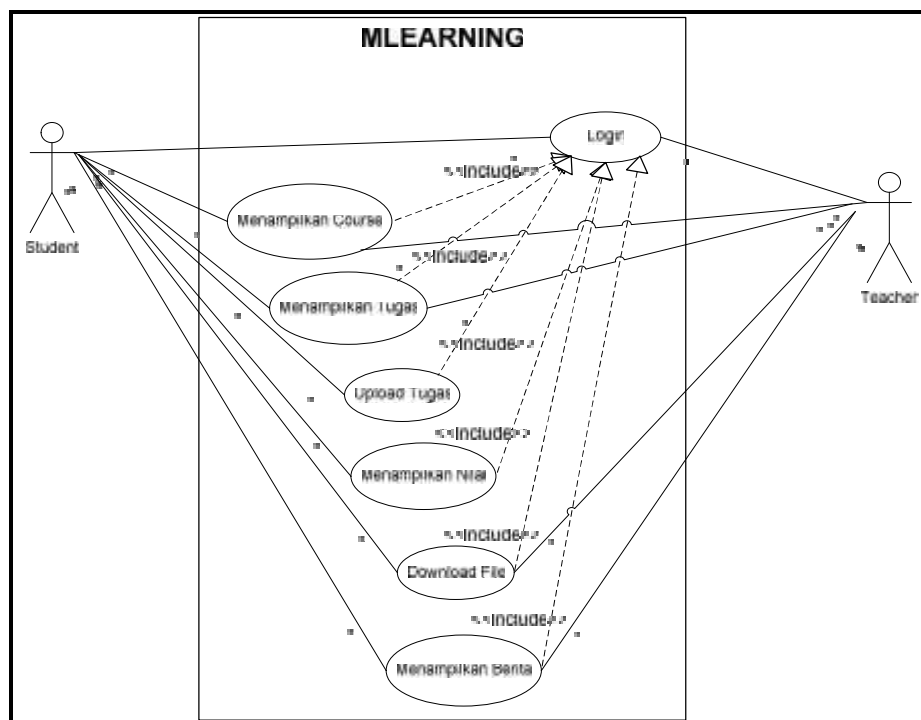
Setelah dilakukan beberapa tahapan dalam analisa sistem, maka dapat dilakukan beberapa perancangan sistem *mobile learning client server* berbasis *Android*. Perancangan-perancangan yang akan dijelaskan dalam laporan ini meliputi perancangan model dalam bentuk UML (*Unified Modeling Language*) yang terdiri dari *Use Case Diagram*, *Class Diagram*, *Activity Diagram*, *Sequence Diagram* dan *Deployment Diagram*. Selain itu juga ada perancangan hubungan antar *class* di *Android* dengan modul penghubung dan perancangan *interface* sistem yang terdiri dari perancangan *prototype* dan struktur menu.

4.5.1. Pemodelan UML (*Unified Modeling Language*)

Analisa perancangan pada sistem ini menggunakan UML, yaitu terdiri dari *use case diagram*, *class diagram*, *activity diagram*, *sequence diagram* dan *deployment diagram*. Untuk lebih jelasnya dapat dilihat pada lampiran A.

4.5.1.1 *Use Case Diagram*

Use case diagram merupakan suatu aktivitas yang menggambarkan urutan interaksi antar satu atau lebih aktor dan sistem. *Use case* yang akan dirancang yaitu *use case diagram* untuk pengaksesan melalui perangkat *Android*. Gambar 4.2 dibawah ini menjelaskan aliran *use case diagram* pengaksesan melalui perangkat *Android*.



Gambar 4.2. Aliran *Use case diagram* (pengaksesan melalui perangkat *Android*)

Dari gambar 4.2 dapat dilihat sistem ini terdiri dari 1 aktor dan 8 *use case*. Untuk lebih jelasnya, Spesifikasi dari *use case diagram* (pengaksesan melalui perangkat *Android*) dapat di lihat pada tabel dibawah ini.

Tabel 4.2. Spesifikasi *use case login*

Aktor Utama	<i>Student</i>
Kondisi Awal	<i>Username</i> dan <i>password</i> masih kosong
Kondisi Akhir	Masuk ke halaman <i>welcome</i> , sebagai halaman utama
Main success scenario	1. <i>Student</i> mengetikkan <i>username</i> dan <i>password</i> 2. Klik tombol <i>login</i>
Exception	Jika <i>username</i> atau <i>password</i> salah, maka akan keluar kotak dialog dengan membawa pesan <i>username</i> atau <i>password</i> salah.

Tabel 4.3. Spesifikasi *use case* menampilkan *course*

Aktor Utama	<i>Student</i>
Kondisi Awal	Halaman utama
Kondisi Akhir	Tampil <i>list</i> detail isi <i>course</i> berdasarkan topik
Main success scenario	1. <i>Student</i> melakukan klik <i>icon course</i> 2. Tampil <i>list course</i> yang telah di <i>enrol</i> oleh <i>student</i> . 3. <i>Student</i> memilih <i>course</i> 4. Tampil <i>list</i> topik yang ada didalam <i>course</i> . 5. <i>Student</i> memilih salah satu topik. 6. Tampil detail isi topik yang dipilih.

Tabel 4.4. Spesifikasi *use case* menampilkan tugas

Aktor Utama	<i>Student</i>
Kondisi Awal	Halaman utama
Kondisi Akhir	Tampil <i>list</i> detail isi tugas berdasarkan <i>course</i>
Main success scenario	1. <i>Student</i> melakukan klik <i>icon</i> tugas. 2. Tampil <i>list course</i> . 3. <i>Student</i> memilih salah satu <i>course</i> yang ada tugas. 4. Tampil <i>list</i> tugas. 5. <i>Student</i> memilih tugas. 6. Tampil detail isi tugas berdasarkan <i>course</i> .

Tabel 4.5. Spesifikasi *use case upload* tugas

Aktor Utama	<i>Student</i>
Kondisi Awal	Halaman utama
Kondisi Akhir	Tampil tombol <i>upload</i> tugas berupa file maupun teks

Main success scenario	<ol style="list-style-type: none"> 1. <i>Student</i> melakukan klik <i>icon</i> tugas. 2. Tampil <i>list course</i>. 3. <i>Student</i> memilih salah satu <i>course</i> yang ada tugas. 4. Tampil <i>list</i> tugas. 5. <i>Student</i> memilih tugas. 6. Tampil tombol <i>upload</i> tugas berupa file maupun teks
------------------------------	--

Tabel 4.6. Spesifikasi *use case* menampilkan nilai

Aktor Utama	<i>Student</i>
Kondisi Awal	Halaman utama
Kondisi Akhir	Tampil nilai berdasarkan tugas
Main success scenario	<ol style="list-style-type: none"> 1. <i>Student</i> melakukan klik <i>icon</i> nilai. 2. Tampil <i>list course</i>. 3. <i>Student</i> memilih salah satu <i>course</i> yang ada tugas. 4. Tampil <i>list</i> tugas. 5. <i>Student</i> memilih tugas. 6. Tampil nilai dari tugas yang dipilih.

Tabel 4.7. Spesifikasi *use case* download file

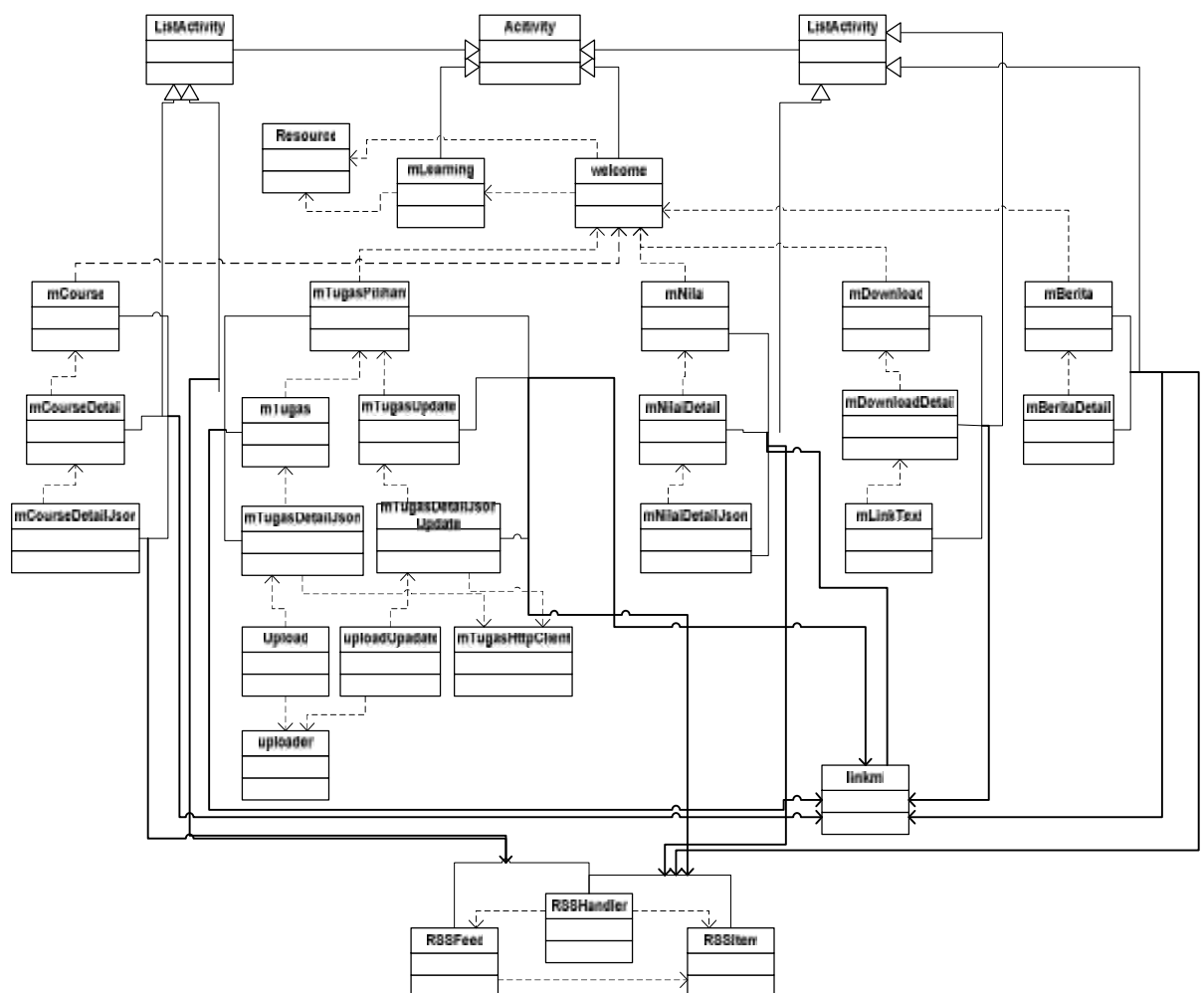
Aktor Utama	<i>Student</i>
Kondisi Awal	Halaman utama
Kondisi Akhir	Tampil halaman <i>link download</i>
Main success scenario	<ol style="list-style-type: none"> 1. <i>Student</i> memilih <i>icon download</i> 2. Tampil <i>list course</i>. 3. <i>Student</i> memilih <i>course</i> yang berisi file <i>download</i>. 4. Tampil <i>list</i> file yang akan <i>download</i>. 5. <i>Student</i> memilih salah satu file. 6. Tampil <i>link download</i>.

Tabel 4.8. Spesifikasi *use case* menampilkan berita

Aktor Utama	<i>Student</i>
Kondisi Awal	Halaman utama
Kondisi Akhir	Tampil halaman berita
Main success scenario	<ol style="list-style-type: none"> 1. <i>Student</i> memilih <i>icon</i> berita 2. Tampil <i>list</i> berita. 3. <i>Student</i> memilih berita. 4. Tampil detail berita.

4.5.1.2. Class Diagram

Class Diagram menggambarkan struktur dan deskripsi *class*, *package*, dan objek yang saling terhubung. *Class diagram* yang dijelaskan pada analisa ini adalah *class diagram* sistem yang terpasang pada perangkat *Android*. Gambar 4.3 dibawah ini menjelaskan *class diagram* sistem yang terpasang pada perangkat *Android*.



Gambar 4.3. Class diagram sistem di perangkat *Android*

Berikut detail masing-masing *class* yang berisi nama *class*, *atribut*, dan *method*, yang dapat di lihat pada tabel 4.9 di bawah ini.

Tabel 4.9. Detail *class diagram*

No	Class	Attribut	Method
1	mAbout	- about : TextView	- onCreate(savedInstanceState : Bundle) : void - onClick(v : View) : void
2	mBerita	- linkurl : linkml - SERVER_URL : String - myRssFeed : RSSFeed	- onCreate(savedInstanceState : Bundle) : void - onItemClick(l : ListView,v : View,position : int,id : long) : void
3	mDownload	- linkurl : linkml - SERVER_URL : String - myRssFeed : RSSFeed	- onCreate(savedInstanceState : Bundle) : void - onItemClick(l : ListView,v : View,position : int,id : long) : void
4	mdownloadDetail	- linkurl : linkml - SERVER_URL : String - myRssFeed : RSSFeed	- onCreate(savedInstanceState : Bundle) : void onItemClick(l : ListView,v : View,position : int,id : long) : void
5	mLinkText	- linkurl : linkml - url : String - SERVER_URL : String	- onCreate(savedInstanceState : Bundle) : void
6	linkml	- urlta : String	- <<create>> linkml(urlml : String) - getUrl() : String
7	mCourse	- linkurl : linkml - SERVER_URL : String - myRssFeed : RSSFeed	- onCreate(savedInstanceState : Bundle) : void - onItemClick(l : ListView,v : View,position : int,id : long) : void
8	mCourseDetail	- linkurl : linkml - SERVER_URL : String - myRssFeed : RSSFeed	- onCreate(savedInstanceState : Bundle) : void - onItemClick(l : ListView,v : View,position : int,id : long) : void
9	mCourseDetailJson	- linkurl : linkml - judul : TextView - isi : TextView - SERVER_URL : String	- onCreate(savedInstanceState : Bundle) : void
10	mLearning	- linkurl : linkml - SERVER_URL : String - login : Button - username : EditText	- onCreate(savedInstanceState : Bundle) : void - tryLogin(mUsername : String,mPassword : String) : String - createDialog(title : String,text : String) : void

Tabel 4.9. Detail *class diagram* (lanjutan)

		<ul style="list-style-type: none"> - password : EditText - progressDialog : ProgressDialog 	
11	mNilai	<ul style="list-style-type: none"> - linkurl : linkml - <i>SERVER_URL</i> : String - myRssFeed : RSSFeed 	<ul style="list-style-type: none"> - onCreate(savedInstanceState : Bundle) : void - onListItemClick(l : ListView,v : View,position : int,id : long) : void
12	mNilaiDetail	<ul style="list-style-type: none"> - linkurl : linkml - <i>SERVER_URL</i> : String - myRssFeed : RSSFeed 	<ul style="list-style-type: none"> - onCreate(savedInstanceState : Bundle) : void - onListItemClick(l : ListView,v : View,position : int,id : long) : void
13	mNilaiDetailJson	<ul style="list-style-type: none"> - linkurl : linkml - judulNilai : TextView - isiNilai : TextView - <i>SERVER_URL</i> : String 	<ul style="list-style-type: none"> - onCreate(savedInstanceState : Bundle) : void
14	RSSFeed	<ul style="list-style-type: none"> - title : String - description : String - link : String - pubDate : String - itemList : List<RSSItem> 	<ul style="list-style-type: none"> - <<create>> RSSFeed() - addItem(item : RSSItem) : void - getItem(location : int) : RSSItem - getList() : List<RSSItem> - setTitle(value : String) : void - setDescription(value : String) : void - setLink(value : String) : void - setPubdate(value : String) : void - getTitle() : String - getDescription() : String - getLink() : String - getPubdate() : String
15	RSSHHandler	<ul style="list-style-type: none"> - state_unknown : int - state_title : int - state_description : int - state_link : int - state_pubdate : int - currentState : int - feed : RSSFeed - item : RSSItem - itemFound : boolean 	<ul style="list-style-type: none"> - <<create>> RSSHandler() - getFeed() : RSSFeed - startDocument() : void - endDocument() : void - startElement(uri : String,localName : String,Name : String,attributes : Attributes) : void - endElement(uri : String,localName : String,qName : String) : void - characters(ch : char[],start : int,length : int) : void
16	RSSItem	<ul style="list-style-type: none"> - title : String - description : String - link : String 	<ul style="list-style-type: none"> - <<create>> RSSItem() - setTitle(value : String) : void - setDescription(value : String) : void

Tabel 4.9. Detail *class diagram* (lanjutan)

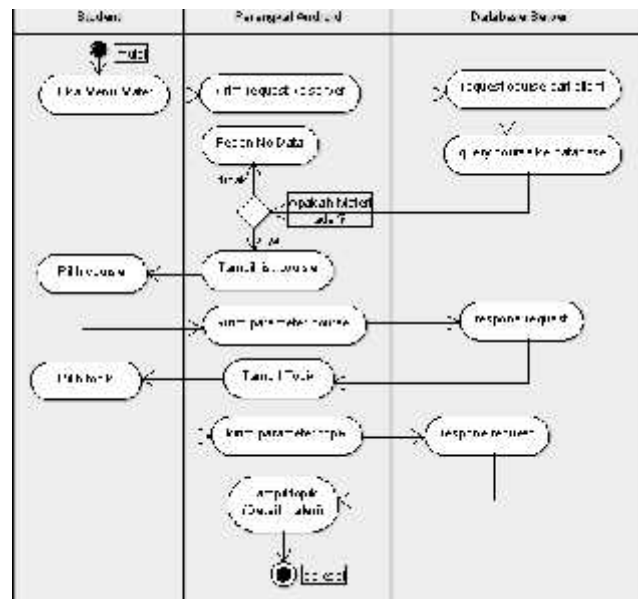
		- pubDate : String	- setLink(value : String) : void - setPubdate(value : String) : void - getTitle() : String - getDescription() : String - getLink() : String - getPubdate() : String - toString() : String
17	mBeritaDetail	-	- onCreate(savedInstanceState : Bundle) : void
18	mTugasPilihan	-	- onCreate(savedInstanceState : Bundle) : void
19	mTugas	- linkurl : linkml - SERVER_URL : String - myRssFeed : RSSFeed	- onCreate(savedInstanceState : Bundle) : void - onItemClick(l : ListView,v : View,position : int,id : long) : void
20	mTugasUpdate	- linkurl : linkml - SERVER_URL : String - myRssFeed : RSSFeed	- onCreate(savedInstanceState : Bundle) : void - onItemClick(l : ListView,v : View,position : int,id : long) : void
21	mTugasDetailUpdate	- linkurl : linkml - SERVER_URL : String - myRssFeed : RSSFeed	- onCreate(savedInstanceState : Bundle) : void - onItemClick(l : ListView,v : View,position : int,id : long) : void
22	mTugasDetail	- linkurl : linkml - SERVER_URL : String - myRssFeed : RSSFeed	- onCreate(savedInstanceState : Bundle) : void - onItemClick(l : ListView,v : View,position : int,id : long) : void
23	mTugasDetailJson	- linkurl : linkml - judulTugas : TextView - isiTugas : TextView - jawaban : EditText - SERVER_URL : String	- onCreate(savedInstanceState : Bundle) : void - createDialog(title : String,text : String) : void
24	mTugasDetailJson Update	- linkurl : linkml - judulTugas : TextView - isiTugas : TextView - jawaban : EditText - SERVER_URL : String	- onCreate(savedInstanceState : Bundle) : void - createDialog(title : String,text : String) : void
25	mTugasHttpClient	- HTTP_TIMEOUT : int - mHttpClient : HttpClient	- getHttpClient() : HttpClient - executeHttpPost(url : String,postParameters : ArrayList<NameValuePair>) : String - executeHttpGet(url : String) : String

Tabel 4.9. Detail *class diagram* (lanjutan)

26	<i>upload</i>	<ul style="list-style-type: none"> - linkurl : linkml - <i>SERVER_URL</i> : String - item : List<String> - path : List<String> - root : String - myPathText : EditText 	<ul style="list-style-type: none"> - onCreate(savedInstanceState : Bundle) : void - getDir(dirPath : String) : void - onListItemClick(l : ListView,v : View,position : int,id : long) : void
27	<i>uploadUpdate</i>	<ul style="list-style-type: none"> - linkurl : linkml - <i>SERVER_URL</i> : String - item : List<String> - path : List<String> - root : String - myPathText : EditText 	<ul style="list-style-type: none"> - onCreate(savedInstanceState : Bundle) : void - getDir(dirPath : String) : void - onListItemClick(l : ListView,v : View,position : int,id : long) : void
28	<i>Uploader</i>	<ul style="list-style-type: none"> - connectURL : URL - params : String - responseString : String - fileName : String - dataToServer : byte[] - fileInputStream : FileInputStream - info : TextView 	<ul style="list-style-type: none"> - setUrlAndFile(urlString : String,fileName : String,info : TextView) : void - doUpload() : void - doInBackground() : Object
29	welcome	<ul style="list-style-type: none"> - ib:ImageButton 	<ul style="list-style-type: none"> - onCreate(savedInstanceState : Bundle) : void

4.5.1.3. Activity Diagram

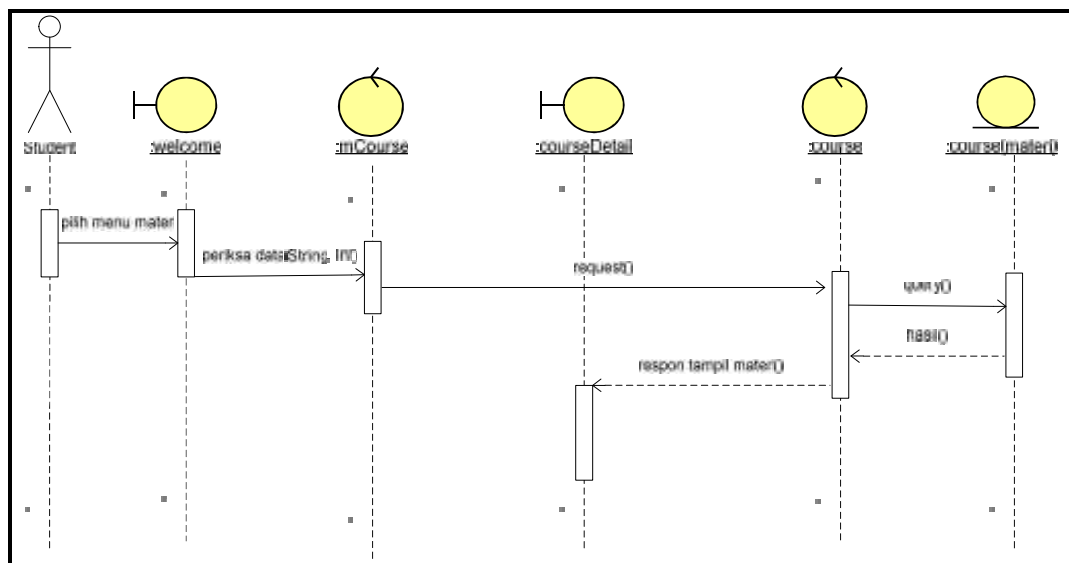
Activity diagram merupakan alur kerja pada setiap use case. *Activity diagram* pada analisa ini mencakup *activity diagram* setiap *use case*. Gambar 4.4. dibawah ini menjelaskan *activity* menampilkan *course* pada perangkat *Android*. Untuk *activity diagram* lainnya dapat di lihat pada lampiran A.



Gambar 4.4. Activity diagram menampilkan course

4.5.1.4. Sequence Diagram

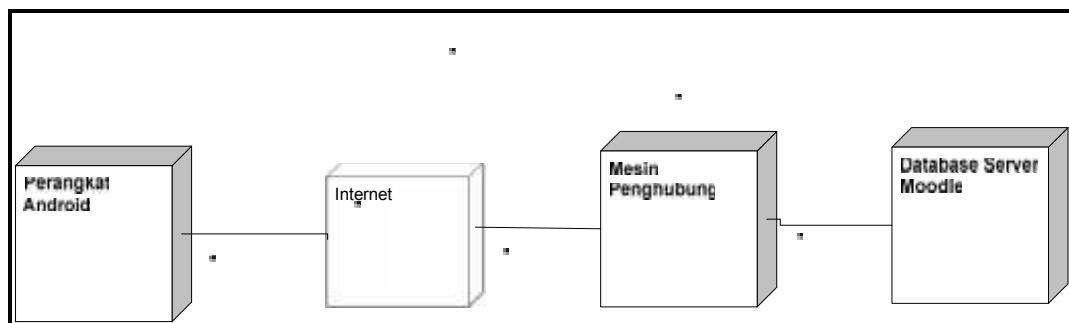
Sequence Diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display* dan sebagainya) berupa pesan yang digambarkan terhadap waktu. Gambar 4.5 dibawah ini menggambarkan *sequence diagram* untuk menampilkan *course* pada perangkat *Android*. Untuk *Sequence diagram* lainnya dapat di lihat pada lampiran A.



Gambar 4.5. Sequence diagram menampilkan course

4.5.1.5. *Deployment Diagram*

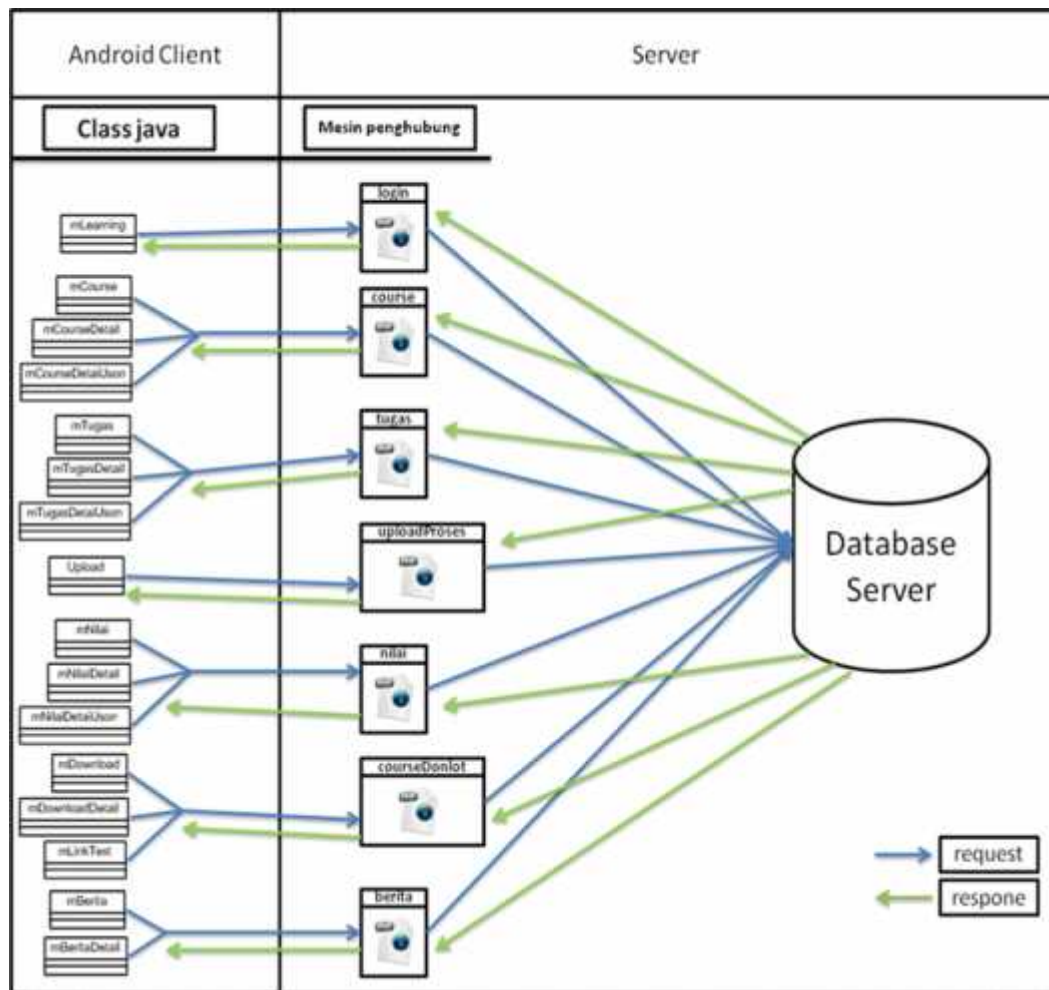
Deployment diagram menggambarkan detail bagaimana komponen dibangun dalam infrastruktur sistem, di mana komponen akan terletak, kemampuan jaringan pada lokasi tersebut, spesifikasi *server*, dan hal-hal lain yang bersifat fisik. Gambar 4.6 dibawah ini menggambarkan hubungan antara *software* dan *hardware* yang digunakan pada sistem informasi yang dibangun.



Gambar 4.6. *Deployment diagram*.

4.5.2. Hubungan *Class* di *Android* dengan Mesin Penghubung

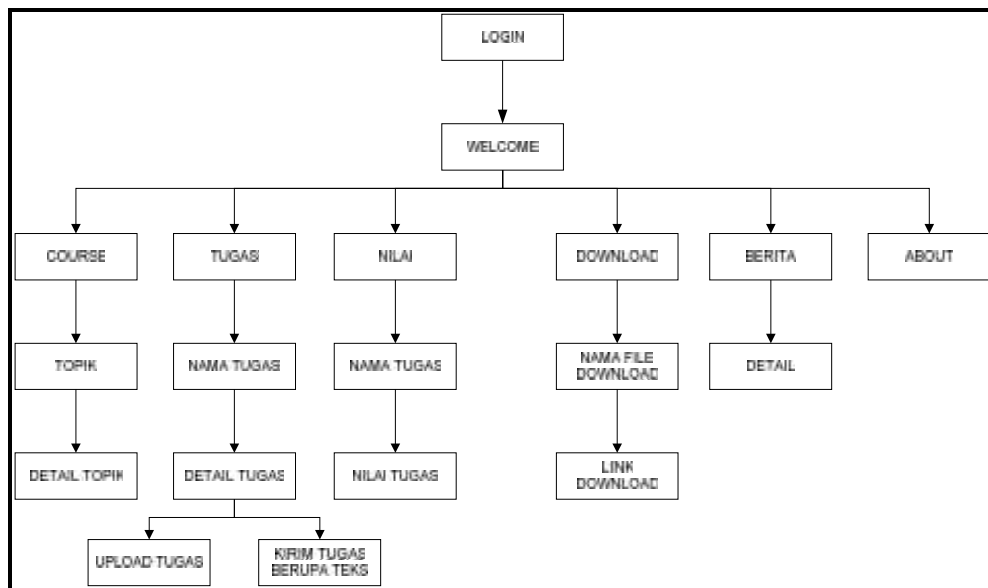
Seperti yang telah dijelaskan pada bagian analisa, untuk mengakses konten *learning* yang ada di *server*, sistem yang berjalan di *Android* membutuhkan bantuan dari luar, yaitu sebuah mesin penghubung. Mesin penghubung berfungsi sebagai jembatan antara sistem di *Android* dengan *server*. Jadi, mesin penghubung ini berperan sebagai perantara *request* dari sistem di *Android* untuk diteruskan ke *server* dan perantara respon dari *server* untuk diteruskan ke sistem di *Android*. Untuk lebih jelasnya bagaimana sistem kerja dari mesin penghubung ini dapat dilihat pada gambar 4.7 dibawah ini.



Gambar 4.7. Hubungan *Class* di *Android* dengan mesin Penghubung

4.5.3. Rancangan Struktur Menu Sistem

Rancangan struktur menu merupakan tahapan untuk merancang bagaimana struktur menu yang akan dibangun. Berikut struktur menu dari sistem yang akan dibangun dapat dilihat pada gambar 4.8 di bawah ini.



Gambar 4.8. Rancangan Struktur menu

4.5.4. Perancangan Antarmuka Pengguna Sistem

Rancangan antar muka pengguna sistem berfungsi sebagai landasan awal dalam merancang tampilan antar muka sistem. Pada analisa dan peracannngan ini, antar muka untuk pengguna sistem pada perangkat *Android* antara lain, antar muka *login*, halaman utama, halaman *course*, halaman tugas, halaman *upload* tugas, halaman nilai, halaman *download*, halaman *about*. Untuk lebih lengkapnya dapat dilihat pada lampiran B.

4.5.4.1 Perancangan antar muka *login* di *Android*.

Gambar 4.9 di bawah ini menjelaskan perancangan antar muka *login* pada perangkat *Android*.

Gambar 4.9. Perancangan antar muka *login* pada perangkat *Android*.

Deskripsi gambar 4.9. tentang perancangan antar muka *login* pada perangkat *Android* dapat dilihat pada tabel 4.10 di bawah ini.

Tabel 4.10. Deskripsi antar muka *login* pada perangkat *Android*.

No	Nama <i>Item</i>	Deskripsi
1.	<i>Mobile learning with Moodle</i>	Merupakan sebuah <i>widget</i> berupa <i>TextView</i>
2.	<i>Username</i>	Sebuah <i>widget EditText</i> yang berfungsi untuk masukan <i>username</i> .
3.	<i>Password</i>	Sebuah <i>widget EditText</i> yang berfungsi untuk masukan <i>password</i> .
4.	<i>Login</i>	Sebuah <i>widget Button</i> yang berfungsi sebagai tombol <i>login</i> .

BAB V

IMPLEMENTASI DAN PENGUJIAN

5.1. Implementasi

Tahapan implementasi merupakan tahapan dimana aplikasi yang telah dirancang, dianalisa, dan dibangun, lalu diuji kelayakannya untuk selanjutnya dioperasikan sebagaimana mestinya sesuai dengan fungsi dan kelayakannya. Berikut ini akan dijelaskan tentang implementasi dari analisis dan perancangan yang telah dilakukan terhadap aplikasi *mobile learning client server* berbasis *moodle* pada *platform Android* ini

5.1.1. Lingkungan Pengembangan

Komponen-komponen yang dibutuhkan untuk mengembangkan aplikasi ini antara lain berupa komponen perangkat keras dan perangkat lunak.

1. Perangkat keras

Processor : *Intel(R) Core(TM)2 Duo CPU T5250 @1.50 GHz*

Memori (RAM) : 1.50 GB

2. Perangkat Lunak

Sistem Operasi : *Windows 7 Profesional 32-bit Operating System*

Bahasa Pemrograman : Java dan PHP

Tools Pengembangan : Eclipse Galileo 3.5, Notepad ++

. : *Java Development Kit 6u24 (JDK 6u24)*

: *Android SDK, ADT 8.0*

: *Android Virtual Device 2.2 (Froyo)*

Browser : *Mozilla Firefox, Chrome*

Server : XAMPP (Apache 2, MySQL, PhpMyAdmin)

Pemodelan UML : Argo UML, Microsoft Visio

5.1.2. Lingkungan Implementasi

Untuk lingkungan implementasi aplikasi ini dilakukan dengan menggunakan perangkat keras dan perangkat lunak diantaranya:

1. Perangkat keras : *Smartphone Android Samsung Galaxy Mini*
2. Perangkat lunak : Sistem operasi *Android 2.2 (froyo)*

Dalam implementasi aplikasi ini, (*Learning Management System*) *LMS moodle* dan *database* dipasang di sebuah penyedia layanan *hosting* dengan *domain* <http://mobile-ta.com/moodle>. untuk mengaksesnya menggunakan *browser Mozilla firefox dan Chrome*.

5.1.3. Tahap-tahap Implementasi

Pada tahap-tahap implementasi ini akan dijelaskan bagaimana tahapan yang penulis lakukan dalam implementasi aplikasi yang telah dibangun. Adapun tahap-tahap yang dilakukan adalah:

5.1.3.1 Instalasi Moodle

Tahap instalasi *Moodle* merupakan tahap pemasangan *Moodle* di *server* yang nantinya berfungsi sebagai *server* konten *mobile learning*. *Moodle* yang digunakan pada tahap instalasi ini adalah versi *moodle 1.9*. Pemilihan versi *Moodle 1.9* ini dikarenakan kestabilannya. Untuk melakukan instalasi *moodle* ada beberapa tahapan yang harus kita lakukan :

1. Ekstrak *file Moodle* yang telah di download dari situs *Moodle*.
2. Letakan *file moodle* di dalam folder *public_html* pada *server*.
3. Akses melalui *browser*, kemudian lakukan instalasi sesuai instruksi.

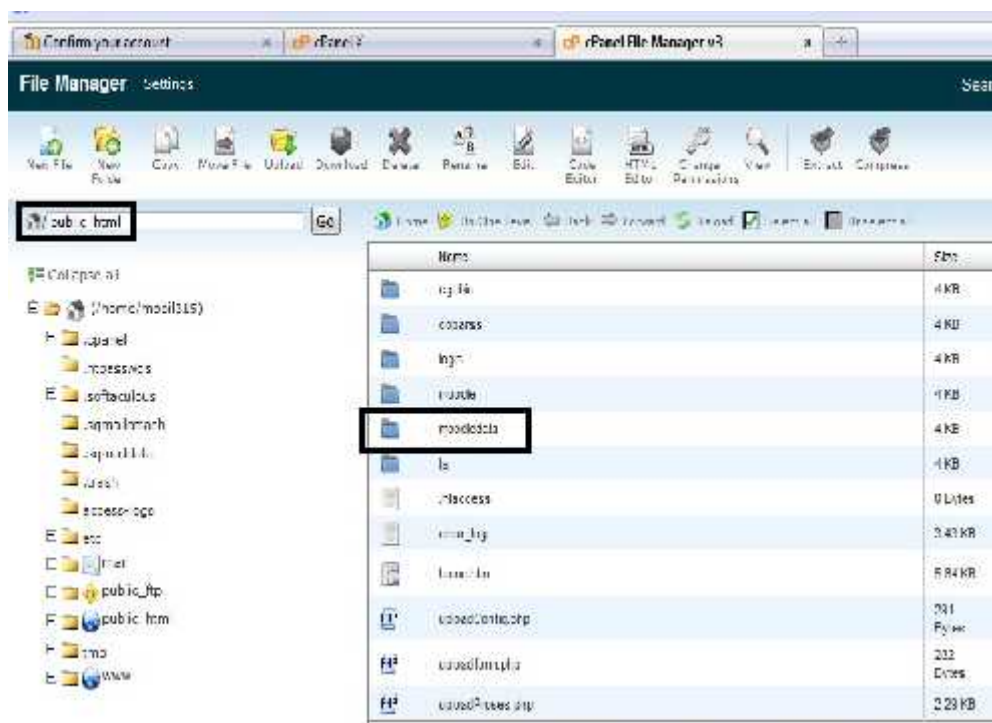
5.1.3.2 Konfigurasi Awal

Tahap konfigurasi awal ini adalah tahap konfigurasi beberapa *file* di *server* yang berhubungan dengan *LMS moodle*, diantaranya:

1. *Data directory Moodle*

Secara *default* saat *install Moodle* di *Server*, maka *data directory* akan otomatis terbuat di *root server*. Untuk bisa implementasi aplikasi *mobile*

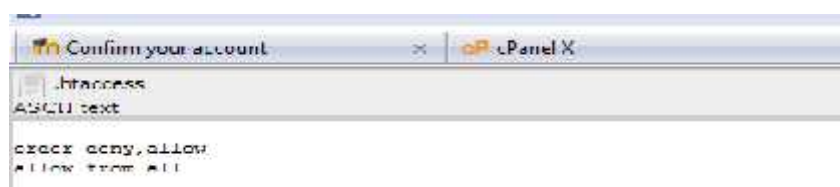
learning ini, data *directory moodle* harus di pindahkan ke dalam *directory public_html* pada *server*. Untuk lebih jelasnya dapat dilihat pada gambar 5.1 di bawah ini.



Gambar 5.1. Letak data *directory Moodle*

2. File *.htaccess*

Agar salah satu menu di aplikasi *mobile learning* nantinya bisa berjalan sesuai yang diharapkan. Maka, diperlukan konfigurasi pada file *.htaccess Moodle*. File ini terletak di dalam data *directory moodle*, tepatnya dalam *directory moodledata*. Hasil akhir konfigurasi file *.htaccess* dapat dilihat pada gambar 5.2 di bawah ini.



Gambar 5.2. Hasil akhir konfigurasi file *.htaccess*.

3. *File config.php*

Konfigurasi *file config.php* ini bertujuan untuk *disable passwordsaltmain* di *Moodle*. Konfigurasi pada tahap ini bersifat optional, jika *passwordsaltmain* tidak *disable* maka aplikasi *mobile learning* di *Android* tidak bisa *login*. Pilihan lain, jika *passwordsaltmain* *enable* maka akan ada konfigurasi tambahan pada *file* mesin penghubung. Jadi pada implementasi *mobile learning* kali ini *passwordsaltmain* di *disable*. Untuk *disable passwordsaltmain*, edit *file config.php* yang terletak di dalam *directory moodle*, kemudian cari baris kode

```
"$CFG->passwordsaltmain='/ BzxI#kpNHLk)ASv.v[ay^c{9BzC;qq';"
```

ubah menjadi `"$CFG->passwordsaltmain = ''"`.

5.1.3.3. Instalasi Mesin Penghubung ke *Hosting*

Seperti yang telah dijelaskan pada bab analisa dan perancangan, mesin penghubung mempunyai peran penting dalam aplikasi *mobile learning client server* berbasis *Android* ini. Mesin ini berperan sebagai penghubung antara aplikasi di *Android* dengan konten *learning* yang ada di *server*, sehingga untuk bisa implementasi aplikasi yang telah dibuat, kita harus meletakkan *Mesin* ini bersamaan dengan *file Moodle* di *hosting*.

File mesin ini diletakan di dalam sebuah *direktory* yang di beri nama *directory "ta"*, kemudian *directory ta* ini di letakan di *directory public_html server*. Untuk lebih jelasnya dapat dilihat pada gambar 5.3 dibawah .

Mesin penghubung sendiri dibangun menggunakan bahasa pemrograman PHP, secara umum *Mesin* ini berisi baris kode yang bekerja meneruskan *query* dan respon dari *client* dan *server*. Mesin 5.1 di bawah ini merupakan mesin *login* yang menangani masalah *login* dan untuk *source code Mesin* lainnya dapat dilihat pada lampiran C


```

<?php
include "config.php";

$username = $_REQUEST['username'];
$password = $_REQUEST['password'];
$passwordhash = md5($password);

// query untuk mendapatkan record dari username
$query = "SELECT * from mdl_user where username = '$username'
and password = '$passwordhash' ";
$numRow = mysql_num_rows(mysql_query($query));

// cek kesesuaian password
if ($numRow >= 1)
echo 1; // for correct login response
else
echo 0; // for incorrect login response
?>

```

Mesin 5.1. mesin login.php

5.1.3.4. Instalasi Aplikasi *Mobile Learning*

Tahap ini merupakan tahap memasang aplikasi *mobile learning* yang telah dibangun berdasarkan analisa dan perancangan. Aplikasi dipasang pada perangkat *Android*. Perangkat *Android* yang digunakan yaitu *smartphone Samsung Galaxy mini*. Untuk melakukan instalasi aplikasi, cukup klik aplikasi *mobile learning* yang telah di *package* ke dalam format *.apk (mLearning.apk) pada perangkat *Android* dan selesai. Aplikasi siap dijalankan.

5.1.4. Implementasi Unjuk Kerja Aplikasi Pada Perangkat *Android*

Implementasi unjuk kerja pada perangkat *Android* merupakan tahap unjuk kerja aplikasi yang telah dibangun langsung di perangkat *Android*. Implementasi kali ini menggunakan *smartphone Samsung Galaxy* dengan sistem operasi *Android 2.2 (Froyo)*. Hasil implementasi menampilkan *course* menggunakan *smartphone Samsung Galaxy Mini* dapat dilihat pada gambar 5.4 di bawah ini. Hasil implementasi lainnya dapat dilihat pada lampiran D.



Gambar 5.4. Hasil implementasi menampilkan *course*.

Sistem yang dibangun berjalan dengan baik di *smartphone Samsung Galaxy Mini*, hal ini dilihat dari keberhasilan sistem mengakses konten *course* dari *server* dan menampilkannya di perangkat *Android*.

Sistem yang dibangun memiliki kelebihan, yaitu pengguna dapat mengakses konten *learning LMS Moodle* dari manapun dengan menggunakan perangkat *Android*, dengan syarat perangkat *Android* harus terhubung dengan internet. Selain kelebihan, sistem ini juga memiliki kekurangan, yaitu belum memiliki fitur yang lengkap sesuai dengan *LMS Moodle*.

5.2. Pengujian Sistem

Tahapan pengujian dilakukan untuk mengetahui apakah sistem yang dibangun telah sesuai dengan yang di harapkan. Pada tahap pengujian sistem ini perangkat keras yang digunakan yaitu *smartphone Samsung Galaxy mini*. Sedangkan material pengujian untuk sistem ini menggunakan data yang telah dimasukan oleh *teacher* atau admin ke database *server*. Pengujian yang akan

dilakukan adalah pengujian akses ke sistem dan pengujian fungsionalitas sistem menggunakan metode *blackbox*

5.2.1. Pengujian Akses Aplikasi *Mobile Learning*

Pengujian yang dilakukan bertujuan untuk mengetahui proses hasil dari sistem, yaitu memperlihatkan aplikasi *mobile learning* yang telah dibangun bisa diakses melalui perangkat *Android*. Hasil dari pengujian dapat dilihat ketika telah melakukan *login* ke sistem dan sistem menampilkan menu dan submenu. Kemudian pembuktian dilakukan untuk membuktikan bahwa konten yang ditampilkan di sistem sesuai dengan *database server*. Untuk alamat *website* yang digunakan sebagai *server mobile learning* untuk *Android* adalah <http://mobile-ta.com/moodle>.

Pada tabel 5.1 di bawah ini menjelaskan pengujian akses ke sistem *mobile learning* menggunakan *smartphone* Samsung Galaxy Mini.

Tabel 5.1 Pengujian akses ke aplikasi *mobile learning* dari *device Android*

Pukul/Tanggal	Perangkat <i>Android</i>	<i>Provider</i>	Pengujian	Hasil
22.10/ 19 September 2011	Samsung Galaxy Mini	Axis	Koneksi ke <i>server</i>	Berhasil
			<i>Login</i>	Berhasil
			<i>Course</i>	Sesuai <i>database</i>
			Topik <i>course</i>	Sesuai <i>database</i>
			Detail isi <i>course</i>	Sesuai <i>database</i>
			Tugas	Sesuai <i>database</i>
			Detail tugas	Sesuai <i>database</i>
			Upload tugas (<i>file</i>)	Berhasil
			Upload tugas (teks)	Berhasil
			Update tugas (<i>file</i>)	Berhasil
			Update tugas (teks)	Berhasil
			Nilai	Sesuai <i>database</i>
			Nilai berdasarkan tugas	Sesuai <i>database</i>
			<i>Download</i>	Sesuai <i>database</i>
			<i>Link download</i>	Sesuai <i>database</i>
			Berita	Sesuai <i>database</i>
19.30/ 22 September 2011	Samsung Galaxy Mini	Telkomsel	Koneksi ke <i>server</i>	Berhasil
			<i>Login</i>	Berhasil
			<i>Course</i>	Sesuai <i>database</i>
			Topik <i>course</i>	Sesuai <i>database</i>
			Detail isi <i>course</i>	Sesuai <i>database</i>
			Tugas	Sesuai <i>database</i>
			Detail tugas	Sesuai <i>database</i>

			Upload tugas (<i>file</i>)	Berhasil
			Upload tugas (teks)	Berhasil
			Update tugas (<i>file</i>)	Berhasil
			Update tugas (teks)	Berhasil
			Nilai	Sesuai <i>database</i>
			Nilai berdasarkan tugas	Sesuai <i>database</i>
			<i>Download</i>	Sesuai <i>database</i>
			<i>Link download</i>	Sesuai <i>database</i>
			Berita	Sesuai <i>database</i>
21.30/ 22 September 2011	Samsung Galaxy Mini	Tri		
			Koneksi ke <i>server</i>	Gagal
			<i>Login</i>	Gagal
			<i>Login</i>	Gagal
			Koneksi ke <i>server</i>	Berhasil
			<i>Login</i>	Berhasil
			<i>Course</i>	Sesuai <i>database</i>
			Topik <i>course</i>	Gagal
			Topik <i>course</i>	Sesuai <i>database</i>
			Detail isi <i>course</i>	Sesuai <i>database</i>
			Tugas	Sesuai <i>database</i>
			Detail tugas	Sesuai <i>database</i>
			Upload tugas (<i>file</i>)	Berhasil
			Upload tugas (teks)	Berhasil
			Update tugas (<i>file</i>)	Berhasil
			Update tugas (teks)	Berhasil
			Nilai	Sesuai <i>database</i>
			Nilai berdasarkan tugas	Sesuai <i>database</i>
			<i>Download</i>	Sesuai <i>database</i>
			<i>Link download</i>	Sesuai <i>database</i>
			Berita	Sesuai <i>database</i>
19.30/ 24 September 2011	Samsung Galaxy Gio	Telkomsel Halo	Koneksi ke <i>server</i>	Berhasil
			<i>Login</i>	Berhasil
			<i>Course</i>	Sesuai <i>database</i>
			Topik <i>course</i>	Sesuai <i>database</i>
			Detail isi <i>course</i>	Sesuai <i>database</i>
			Tugas	Sesuai <i>database</i>
			Detail tugas	Sesuai <i>database</i>
			Upload tugas (<i>file</i>)	Berhasil
			Update tugas (<i>file</i>)	Berhasil
			Update tugas (teks)	Berhasil
			Upload tugas (teks)	Berhasil
			Nilai	Sesuai <i>database</i>
			Nilai berdasarkan tugas	Sesuai <i>database</i>
			<i>Download</i>	Sesuai <i>database</i>
			<i>Link download</i>	Sesuai <i>database</i>
			Berita	Sesuai <i>database</i>

5.2.2. Pengujian *Blackbox* Aplikasi *Mobile Learning*

Pengujian sistem dilakukan untuk memeriksa kekompakan atau kinerja antar komponen sistem yang diimplementasikan. Tujuan utama dari pengujian sistem adalah untuk memastikan bahwa elemen-elemen atau komponen-komponen dari sistem telah berfungsi sesuai dengan yang diharapkan. Salah satu metode pengujian jenis ini dikenal dengan pengujian *blackbox*. Adapun hasil dari pengujian ini dapat dilihat pada tabel 5.2.

Table 5.2. Pengujian aplikasi dengan metode *Blackbox*

No.	Nama Pengujian	Deskripsi	Perangkat <i>Android</i>	Prekondisi	Prosedur Pengujian	Masukan	Keluaran yang diharapkan	Hasil yang Didapat	Kesimpulan
1.	Pengujian <i>Login</i>	Pengujian <i>login</i> dengan memasukkan <i>username</i> dan <i>password</i> yang benar	Samsung Galaxy Mini	<i>Form login</i> telah tampil	Ketik <i>username</i> dan <i>password</i>	<i>Username</i> “harjay” <i>password</i> “harjay_TIF07”	Tampil halaman menu utama.	Tampil halaman menu utama.	Benar
		Pengujian <i>login</i> dengan memasukkan <i>username</i> atau <i>password</i> yang salah	Samsung Galaxy Mini			<i>Username</i> “harja” <i>password</i> “harjay_TIF05”	Tampil kotak dialog dengan pesan “ <i>username</i> atau <i>password</i> salah”	Tampil kotak dialog dengan pesan “ <i>username</i> atau <i>password</i> salah”	Benar
		Pengujian <i>login</i> dengan <i>username</i> <i>password</i> yang kosong	Samsung Galaxy Mini		Kosong kan <i>username</i> dan <i>password</i>	-	Tampil kotak dialog dengan pesan “ <i>username</i> atau <i>password</i> salah”	Tampil kotak dialog dengan pesan “ <i>username</i> atau <i>password</i> salah”	benar
2	Pengujian menampilkan <i>course</i>	Pengujian untuk menampilkan <i>course</i> yang telah di- <i>enrol</i> oleh pengguna.	Samsung Galaxy Mini	Halaman menu utama telah tampil	Klik gambar <i>course</i>	-	Tampil <i>list course</i> yang telah di- <i>enrol</i>	Tampil <i>list course</i> yang telah di- <i>enrol</i>	Benar

Table 5.2. Pengujian aplikasi dengan metode *Blackbox* (lanjutan)

3	Pengujian menampilkan topik <i>course</i>	Pengujian untuk menampilkan topik yang ada di dalam masing-masing <i>course</i>	Samsung Galaxy Mini	Halaman <i>list course</i> telah tampil	Klik salah satu <i>course</i>	-	Tampil <i>list</i> topik yang ada di dalam <i>course</i> yang dipilih	Tampil <i>list</i> topik yang ada di dalam <i>course</i> yang dipilih	Benar
4	Pengujian menampilkan isi topik	Pengujian untuk menampilkan isi topik secara rinci	Samsung Galaxy Mini	Halaman <i>list</i> topik telah tampil	Klik salah satu topik	-	Tampil detail isi dari topik yang di pilih.	Tampil detail isi dari topik yang di pilih.	Benar
5	Pengujian tugas	Pengujian untuk menampilkan tugas berdasarkan <i>course</i>	Samsung Galaxy Mini	Halaman <i>list course</i> telah tampil	Klik salah satu <i>course</i>	-	Tampil <i>list</i> tugas berdasarkan <i>course</i>	Tampil <i>list</i> tugas berdasarkan <i>course</i>	Benar
6	Pengujian detail tugas	Pengujian untuk menampilkan detail tugas yang dipilih	Samsung Galaxy Mini	Halaman <i>list</i> tugas telah tampil	Klik salah satu tugas.	-	Tampil halaman detail tugas yang berisi perintah atau soal atau pertanyaan	Tampil halaman detail tugas yang berisi perintah atau soal atau pertanyaan	Benar

Table 5.2. Pengujian aplikasi dengan metode *Blackbox* (lanjutan)

7	Pengujian upload tugas	Pengujian untuk tugas berupa <i>file</i>	Samsung Galaxy Mini	Halaman detail tugas telah tampil	Klik tombol <i>upload</i> tugas file dan tampil halaman <i>upload</i> , kemudian isikan letak alamat file di perangkat <i>Android</i>	/mnt/sdcard/download/enkrip.pdf	Tampil pesan file berhasil di <i>upload</i>	Tampil pesan file berhasil di <i>upload</i>	Benar
		Pengujian untuk kirim tugas berupa teks			Ketikan jawaban tugas di dalam kotak jawaban.	Jawaban dari pertanyaan yang ada di dalam detail tugas	Tampil pesan “tugas berhasil dikirim”	Tampil pesan “tugas berhasil dikirim”	Benar
8	Pengujian nilai	Pengujian untuk menampilkan nilai berdasarkan tugas	Samsung Galaxy Mini	Halaman list tugas telah tampil	Klik salah satu tugas yang telah diselesaikan.	-	Tampil nilai berdasarkan tugas	Tampil nilai berdasarkan tugas	Benar
9	Pengujian <i>download</i>	Pengujian untuk menampilkan <i>list download file</i>	Samsung galaxy mini	Halaman list <i>course</i> telah tampil.	Klik salah satu <i>course</i>	-	tampil <i>list file</i> yang bisa di <i>download</i>	tampil <i>list file</i> yang bisa di <i>download</i>	Benar

Table 5.2. Pengujian aplikasi dengan metode *Blackbox* (lanjutan)

	Pengujian <i>download</i>	Pengujian untuk menampilkan <i>link download</i>	Samsung galaxy mini	Halaman <i>list file download</i> telah tampil	Klik salah satu <i>file</i> yang akan di <i>download</i>	-	Tampil halaman yang berisi <i>link download</i>	Tampil halaman yang berisi <i>link download</i>	Benar
		Pengujian untuk melakukan <i>download file</i>		Halaman <i>link download</i> telah tampil	Klik <i>link download</i>	-	File selesai di- <i>download</i>	File selesai di- <i>download</i>	Benar
10	Pengujian berita	Pengujian untuk menampillkan berita	Samsung galaxy mini	Halaman <i>list</i> berita telah tampil	Klik salah satu berita	-	Tampil detail berita	Tampil detail berita	Benar

5.2.3 Kesimpulan Pengujian

Setelah dilakukan beberapa pengujian terhadap aplikasi yang telah dibangun, maka dapat ditarik kesimpulan dari hasil pengujian tersebut. Berikut kesimpulannya:

1. Aplikasi *mobile learning* yang dibangun untuk perangkat *Android*, dapat melakukan koneksi ke *server* dan dapat menampilkan konten *learning* sesuai dengan analisa dan perancangan.
2. Aplikasi yang dijalankan di beberapa perangkat *Android* yang berbeda merk dapat berjalan dengan baik.
3. Konten *mobile learning* yang ditampilkan di perangkat *Android* telah sesuai dengan *database* yang di-*server*.
4. Untuk beberapa pengujian yang mengalami kegagalan dalam koneksi ke server disebabkan beberapa kemungkinan. Diantaranya, koneksi jaringan internet yang kurang stabil, *server* belum *up* atau respon yang terlalu lama.
5. Untuk pengembangan lebih lanjut agar ditambahkan fitur *enrol course*, guna mempermudah mendapatkan konten pembelajaran.

BAB VI

PENUTUP

6.1 Kesimpulan

Setelah menyelesaikan serangkaian tahapan terhadap pembangunan rancang bangun aplikasi *mobile learning client server* berbasis *Moodle* pada *Platform Android* yang dimulai dari pengumpulan data tentang *mobile learning*, *Moodle* dan teknologi *Android* hingga pada tahapan pengujian, maka dapat diambil beberapa kesimpulan diantaranya adalah sebagai berikut:

1. Secara umum aplikasi yang dibangun telah bisa berjalan pada perangkat *Android* dan bisa mengakses konten yang ada di *server Moodle* melalui mesin penghubung. Selain itu, aplikasi yang berjalan di *Android* telah bisa memanipulasi *database moodle* dalam bentuk aksi *update*.
2. Dari hasil pengujian yang dilakukan pada kondisi tertentu, sistem di *Android* tidak bisa mengakses *server* dikarenakan adanya gangguan teknis diluar sistem seperti, gangguan jaringan, *server* lambat dalam respon, atau padatnya jaringan, bukan karena sistem di *Android error*.
3. Kekurangan aplikasi ini masih belum bisa untuk melakukann *enrol course* dari perangkat *Android*, belum lengkapnya fitur yang tersedia, dan aspek keamanan dari sisi *client*.

6.2 Saran

Beberapa hal yang disarankan dalam pengembangan aplikasi *mobile learning client server* berbasis *Android* ini adalah sebagai berikut:

1. Pada pengembangan selanjutnya diharapkan fitur yang ditampilkan pada perangkat *Android* bisa lebih lengkap.
2. Pada pengembangan lebih lanjut, diharapkan adanya *session* dari sisi client dan file *.htaccess* bisa di aktifkan.
3. Pada pengembangan selanjutnya diharapkan dapat melakukan *enrol course* dari perangkat *Android*.
4. Pada pengembangan selanjutnya diharapkan ada fitur pemberitahuan di sistem pada perangkat *Android* jika ada konten terbaru yang ditambahkan pada *server*.

DAFTAR PUSTAKA

- Abelson, W.Frank, Robi Sen, Chris King. *Android In Action, Secend Edition*. Manning Publication. 2011.
- Amiral, Muhammad. *Aplikasi Pengingat Sholat dan Arah Kiblat Menggunakan Global Positioning System (GPS) berbasis Android 1.6*. Teknik Informatika, Institut Teknologi Indonesia. 2010.
- Ariyana, Yoki. *M-Learning Sebagai Sumber Belajar berbasis Wireless Application Protol (WAP)*. [Online] Available www.p4tkipa.org/data/mlearn.pdf .11 April 2011.
- Darwiyanti, Sri dan Romi Satria Wahono. *Pengenalan Unified Modeling Language (UML)* .[Online] Available <http://ilmukoputer.org/2006/08/05/pengantar-uml/> 14 April 2011 .
- Effendi, Empy dan Hartono Zhuang. *e-Learning Konsep dan Aplikasi*. Penerbit Andi, Yogyakarta, 2005.
- Kadir, Abdul. *Mastering Ajax dan PHP*. Penerbit Andi, Yogyakarta. 2009.
- Kruchten, Philippe. *The Rational Unified Process An Introduction, Second Edition*. Addison Wesley, 2000.
- Meier, Reto. *Professional Android Application Development*. Wiley Publishing. 2009.
- Mulyadi. *Membuat Aplikasi Untuk Android*. Penerbit Multimedia Center Publishing, Yogyakarta, 2010.
- Murphy, Mark.L. *Beginning Android 2*. Apress. 2010.
- Naughton, Patrick. *Konsep Dasar Pemrograman Java*. Penerbit Andi, Yogyakarta. 1997.
- Parsons, David. *Mobile Learning*. [Online] Available <http://www.massey.ac.nz/~dpparson/746.pdf> 7 April 2011

- Parsons, David, et al. *A Design Requirements Framework for Mobile Learning Environments*. Journal Of Computers, VOL. 2, NO. 4, June 2007.
- Rational Team. *“Rational Unified Process : Best Practices for Software Development Teams”*. 2001.
- Safaat, Nazruddin. *Android: Pemrograman Aplikasi Mobile Smartphone dan Tablet PC*. Penerbit Informatika, Bandung, 2011.
- Sanjaya, Suwanto. *Perancangan dan Implentasi Sistem Informasi Akademik Dengan Teknologi J2ME Menggunakan Jaringan GPRS*. Laporan Tugas Akhir Sarjana, Jurusan Teknik Informatika, Universitas Islam Negeri Sultan Syarif Kasim Riau. 2009.
- Siregar, Ivan Michael, dkk. *Mengembangkan Aplikasi Enterprise Berbasis Android*. Penerbit Gava Media, Yogyakarta, 2010.
- Suhendar, A. *Teknologi Pemograman Mobile Commerce*. Penerbit Informatika, Bandung, 2002.
- Suprianto, Dodit. *Buku Pintar Pemrograman PHP*. OASE Media, Bandung. 2008.
- Tomy. *Tip dan Trik Profesional MySQL 5*. Elex Media Komputindo, Jakarta. 2008
- Vogel, Lars. 7 Juli 2009. Android Development Tutorial. [Online] available. <http://www.vogella.de/articles/Android/article.html>, 30 Juni 2011.
- Wahono, Romi Satria. *Pengantar e-Learning dan Pengembangannya*. [Online] Available <http://ilmukomputer.org/2008/11/25/pengantar-elearning-dan-pengembangannya/>. 13 April 2011.
- Widhiartha, Putu A. *Memahami Lebih Lanjut Tentang e-Learning*. [Online] Available http://www.ilmukomputer.org/wpcontent/uploads/2008/07/widhiartha_elearning.pdf. 13 April 2011.
- _____. Android basic. [Online] <http://developer.android.com/guide/basics/what-is-android.html>. 14 April 2011.
- _____. Moodle Docs. [Online] <http://docs.moodle.org/>. 13 April 2011.
- _____. Code Snippets for android.[Online] <http://anddev.org/>. 2 Juli 2011.